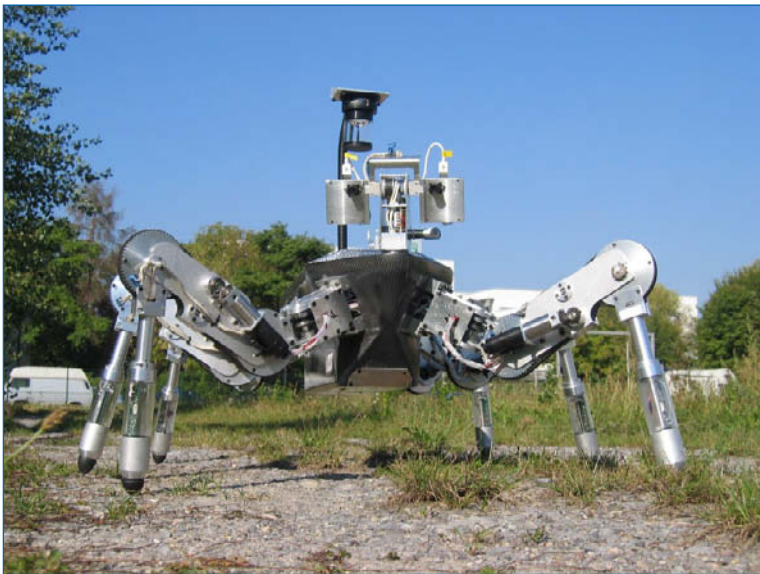


Bernd Gaßmann

Modellbasierte, sensorgestützte Navigation von Laufmaschinen im Gelände

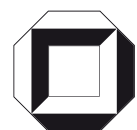


Bernd Gaßmann

**Modellbasierte, sensorgestützte Navigation von Laufmaschinen
im Gelände**

Modellbasierte, sensorgestützte Navigation von Laufmaschinen im Gelände

von
Bernd Gaßmann



universitätsverlag karlsruhe

Dissertation, Universität Karlsruhe (TH)
Fakultät für Informatik, 2007

Impressum

Universitätsverlag Karlsruhe
c/o Universitätsbibliothek
Straße am Forum 2
D-76131 Karlsruhe
www.uvka.de



Dieses Werk ist unter folgender Creative Commons-Lizenz
lizenziert: <http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Universitätsverlag Karlsruhe 2007
Print on Demand

ISBN: 978-3-86644-119-4

Modellbasierte, sensorgestützte Navigation von Laufmaschinen im Gelände

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
von der Fakultät für Informatik
der Universität Fridericiana zu Karlsruhe (TH)
genehmigte

Dissertation

von

Bernd Gaßmann

aus Sinsheim.

Tag der mündlichen Prüfung:	23.01.2007
Erster Gutachter:	Prof. Dr.-Ing. Rüdiger Dillmann
Zweiter Gutachter:	Prof. Dr. rer. nat. Karsten Berns

Vorwort

Als ich mich nach dem Abitur intensiv mit der Frage nach dem geeigneten Studium beschäftigte, war die Informatik nur eine von vielen Alternativen. Da meine Erfahrung im Umgang mit Computern begrenzt war, wollte ich mich über die Inhalte des Informatikstudiums in Karlsruhe aus erster Hand informieren: Michael Tagscherer, der Schwager meines guten Freundes Uwe Friederich verfasste gerade seine Diplomarbeit auf diesem Gebiet: „Adaptive Umweltklassifikation für eine Laufmaschine“ (Tagscherer, 1994). Hierbei ging es um die Steuerung einer sechsbeinigen Laufmaschine namens LAURON; die Faszination der Robotik hatte mich erfasst.

Bei der Ausschau nach einer Diplomarbeit im Bereich Robotik suchte ich gegen Ende meines Studiums Karsten Berns auf, damals Abteilungsleiter der Gruppe *IDS* am *Forschungszentrum Informatik (FZI)*. Das Angebot seines Mitarbeiters Kay-Ulrich Scholl, für LAURON III eine Verhaltenssteuerung zu entwickeln, konnte ich nicht ausschlagen. Der Sechsbeiner hatte mich endgültig in seinen Bann gezogen.

LAURON hat mein Leben nicht zuletzt durch die vorliegende Arbeit in den letzten 13 Jahren entscheidend beeinflusst. Doch was ist schon ein Roboter, ohne die vielen engagierten Menschen, die ihren kreativen Geist und ihre technischen Fertigkeiten zum Erfolg eines derartigen Projektes beisteuern. Ihnen allen gilt mein Dank.

Die ersten Berührungängste mit der für mich als Informatiker doch etwas fremden Mechatronik nahmen mir unsere Elektroingenieure Volker Kepplin und Carsten Hillenbrand; insbesondere für die fruchtbaren Diskussionen hinsichtlich der Navigationsensorik möchte ich mich bedanken.

Wesentliche Beiträge zum Gelingen meiner Arbeit wurden von meinen Studenten erbracht; hierfür danke ich Ihnen. Besonderen Lob möchte ich Lutz Frommberger, Franziska Zacharias, Marco Huber und Tobias Bär für ihre außerordentliche Kreativität aussprechen. Während des Großteils meiner Zeit am *FZI* konnte ich stets auf die großartige Unterstützung von Tobias Bär zählen, vielen Dank.

Für die in allen Lebenslagen immer verfügbare Systemunterstützung – auch im freien Feld – geht mein Dank an die Adresse der Administratoren Jan Albiez, Uli Hochholtinger und Malte Cornils. Als Experte auf dem Gebiet von Laufmaschinen gebührt Jan Albiez weiterer Dank; er stand immer für ein kritisches Gespräch und gute Ratschläge zur Verfügung.

Schon während meiner Diplomarbeit hat mich Kay-Ulrich Scholl unter seine Fittiche genommen. Es war für mich eine große Bereicherung, mit ihm zusammen Industrieprojekte erfolgreich abzuwickeln und das *MCA2*-Framework, mit dem nahezu alle Roboter in unserer Gruppe gesteuert werden, an die unterschiedlichsten Erfordernisse anzupas-

sen. Er war mir während unserer gemeinsamen Zeit am *FZI* ein sehr guter Mentor; herzlichen Dank.

Eine außerordentliche Gemeinschaftsleistung unserer Gruppe mit großem Wert für meine Arbeit stellt die Entwicklung von LAURON IV dar. Unserem Abteilungsleiter Marius Zöllner danke ich für das engagierte Forcieren dieses Projektes und dem nötigen Freiraum, meine Vorstellungen von Navigation umsetzen zu können. Großem Dank bin darüber hinaus meinem Zimmerkollegen Thilo Kerscher verpflichtet: ohne dessen Konstruktion, Planung, sorgfältige Durchführung und unermüdliche, selbstlose Einsatzbereitschaft wäre die Realisierung des Roboters schlichtweg nicht möglich gewesen.

Neben LAURON werden in unserer Gruppe viele größere und kleinere Projekte gemeistert. Wird es an der ein oder anderen Stelle eng, so kann man sich immer voll auf die Mithilfe der Kollegen verlassen. Auch unter steigendem Projektdruck gelingt es der Gruppe, eine freundschaftliche Atmosphäre aufrecht zu erhalten, was insbesondere auch die gemeinsamen Reisen sehr angenehm macht. An dieser Stelle geht mein Dank an Jan Albiez, Tamim Asfour, Clemens Birkenhofer, Michael Deck, Patrick Feucht, Olaf Fischer, Thomas Gump, Carsten Hillenbrand, Winfried Ilg, Volker Kepplin, Thilo Kerscher, Duc Nguyen Ly, Kristian Regenstein, Diana Reiter, Kay-Ulrich Scholl, Klaus Uhl, Zhixing Xue, Marco Ziegenmeyer, Marius Zöllner, allen Studenten der Gruppe *IDS*, sowie allen Mitarbeitern der Uni-Gruppe am *IAIM*.

Für die Anmerkungen und Korrekturen dieser Arbeit möchte ich mich recht herzlich bei Matthias Belz, Tamim Asfour sowie meinem Bruder Andreas bedanken.

Prof. Karsten Berns bin ich zutiefst dankbar, dass er mir als Korreferent beiseite stand. Als Abteilungsleiter am *FZI* hat er die Gruppe *IDS* aufgebaut und war einer der Schöpfer von LAURON. Sein visionäres Denken haben meine Arbeit von Anfang an inspiriert. Ich denke noch gerne an die Tage vor seinem Ruf nach Kaiserslautern zurück, an denen er mich beiseite nahm und bei einer Tasse Kaffee die Grundsteine für meine Arbeit gelegt hat.

Mein besonderer Dank gilt Prof. Rüdiger Dillmann als meinen Doktorvater. Ohne seine fortwährende Unterstützung wäre diese Arbeit nicht möglich gewesen. Sein Engagement und seine Anregungen insbesondere in der Endphase waren sehr wertvoll.

Zuletzt möchte ich meinem Bruder Andreas, meiner Mutter Elfriede und meinem Vater Peter von ganzem Herzen danken. Auf meinem Weg haben sie mich stets begleitet und in meinen Entscheidungen bekräftigt. Sie haben mich uneingeschränkt unterstützt und mir dabei geholfen, das Ziel nicht aus den Augen zu verlieren.

Meine Frau Michaela sowie meine Kinder Tom und Moritz mussten insbesondere in der Endphase dieser Arbeit immer wieder Einschränkungen hinnehmen. Ihre Liebe hat mir immer wieder Kraft und Mut gegeben, um die Anstrengungen zu meistern. Sie sind immer hinter mir gestanden, dafür Liebe ich sie.

Eppingen, im Februar 2007

Inhaltsverzeichnis

1	Einleitung	1
1.1	Navigation	2
1.2	Ziel der Arbeit	3
1.3	Beitrag der Arbeit	4
1.4	Aufbau der Arbeit	5
2	Navigation mobiler Roboter	7
2.1	Allgemeine Anforderungen und Randbedingungen	7
2.2	Navigationsfähigkeiten von Laufmaschinen	8
2.2.1	Adaptive Suspension Vehicle	9
2.2.2	Ambler	10
2.2.3	DANTE II	11
2.2.4	JROB	13
2.2.5	RHex	14
2.2.6	Scorpion	15
2.3	Radgetriebene und hybride Roboter	15
2.3.1	WorkPartner	16
2.3.2	Mars Rover	17
2.4	Zusammenfassung und Schlussfolgerungen	18
3	Lokalisation	21
3.1	Anforderungen und Randbedingungen	21
3.1.1	Besondere Anforderungen in unstrukturiertem Gelände	22
3.1.2	Zusätzliche Anforderungen bei Laufmaschinen	22
3.2	Relative und absolute Verfahren	23
3.2.1	Koppelnavigation	24
3.2.1.1	Odometrie	24
3.2.1.2	Inertiale Navigationssysteme	26
3.2.1.3	Zusammenfassung	28
3.2.2	Absolute Orientierungsbestimmung	28
3.2.2.1	Gravitation	28
3.2.2.2	Sonnenstandsensoren	29
3.2.2.3	Kompass	29
3.2.2.4	Zusammenfassung	29
3.2.3	Aktive Leuchtfeuer	29

3.2.3.1	GPS	30
3.2.4	Landmarken- und kartenbasierte Verfahren / SLAM	32
3.3	Schlussfolgerungen	35
3.4	Odometrie von Laufmaschinen	38
3.4.1	Odometrieberechnung	38
3.4.2	Gewichtung der Beine	40
3.5	Das Kalman-Filter	43
3.5.1	Formulierung des diskreten Kalman-Filters	44
3.5.2	Kalman-Filter-Algorithmus	45
3.6	Implementierungsaspekte des Kalman-Filters	46
3.6.1	Direkte und indirekte Kalman-Filter-Formulierung	46
3.6.2	Formulierung mit und ohne Rückkopplung	47
3.6.3	Schwach und stark gekoppelte INS-GPS-Integration	48
3.7	Das realisierte Kalman-Filter	49
3.7.1	Fortschreibung der Lage	49
3.7.2	Das nichtlineare Kalman-Filter-Modell	51
3.7.2.1	Herleitung der Positionsfehlerdifferentialgleichung	52
3.7.2.2	Herleitung der Orientierungsfehlerdifferentialgleichung	53
3.7.2.3	Zusammenfassung in einer Fehlerdifferentialgleichung	54
3.7.3	Diskretes Kalman-Filter-Modell	54
3.7.4	Besondere Eigenschaften des entwickelten Systemmodells	55
3.8	Ablauf des Lokalisationsalgorithmus	55
3.8.1	Odometrie	55
3.8.2	Prädiktionsschritt	55
3.8.3	Positionskorrektur durch GPS	57
3.8.4	Orientierungskorrektur durch Kompass & Gravitation	58
3.9	Zusammenfassung	58
4	Umweltmodellierung für Laufmaschinen	61
4.1	Anforderungen und Randbedingungen	61
4.1.1	Einschränkungen	62
4.1.2	Besondere Anforderungen in unstrukturiertem Gelände	62
4.1.3	Zusätzliche Anforderungen bei Laufmaschinen	63
4.2	Geometrische und topologische Karten	64
4.3	Belegtheitsgitter	66
4.4	Erste Schlussfolgerungen	69
4.5	Dimensionsaspekte	71
4.6	Die Repräsentation der Umwelt	71
4.6.1	Datenkapselung mit Hilfe eines Schichtenmodell	73
4.6.2	Speicherung in einer Octree- und Quadtree-Datenstruktur	75
4.7	Die dreidimensionale Urkarte	77
4.7.1	Das erweiterte Inferenz-Gitter	77
4.7.2	Die Gitterzelle	78
4.7.3	Einfügen neuer Sensormessungen	79

4.7.4	Realisierung der Gütefunktion	81
4.8	Globale 2½D-Karten	83
4.9	Zusammenfassung	85
5	Bahnplanung	87
5.1	Anforderungen und Randbedingungen	87
5.1.1	Besondere Anforderungen in unstrukturiertem Gelände	88
5.1.2	Zusätzliche Anforderungen bei Laufmaschinen	89
5.1.3	Das Umweltmodell als Planungsgrundlage	89
5.2	Planungsverfahren	90
5.2.1	Distance Transform (Entfernungstransformation)	90
5.2.2	Dekompositionsverfahren	90
5.2.3	Lernverfahren	91
5.2.3.1	Erklärungsbasiertes Lernen	91
5.2.3.2	Evolutionäre Algorithmen	92
5.2.4	Potenzialfeldverfahren	93
5.3	Erste Schlussfolgerungen	94
5.4	Bahnplanung für Laufmaschinen	96
5.4.1	Chen und Kumar (1996)	96
5.4.2	Bai, Low und Teo (2002)	97
5.4.3	Eldershaw und Yim (2001)	100
5.5	Generisches Laufmaschinenmodell	102
5.5.1	Zentralkörpermodell	102
5.5.2	Beinmodell	104
5.6	Bewertungsfunktion	105
5.6.1	Stellungsmachbarkeit A_p	106
5.6.2	Geländezugänglichkeit A_t	107
5.6.3	Geländekomplexität \overline{A}_t	110
5.6.4	3D-Geländekomplexität \widetilde{A}_t	111
5.7	Potenzialfeldverfahren	112
5.8	Abschließender Vergleich	114
5.9	Zusammenfassung	114
6	Adaption an das Gelände	117
6.1	Bewertung von Gelände	117
6.1.1	Bewertung durch den Menschen	117
6.1.2	Maschinelle Bewertung	119
6.2	Schlussfolgerungen	120
6.3	Beschreibung von Gelände	121
6.4	Auswahl von Laufparametern	122
6.4.1	Laufparameter	122
6.4.2	Unscharfe Logik zur Parameterwahl	124
6.5	Dediziertes Setzen von Fußpunkten	125
6.6	Zusammenfassung	126

7	Realisierung, Experimente und Ergebnisse	129
7.1	Die sechsbeinige Laufmaschine LAURON IV	129
7.1.1	Technischer Aufbau	131
7.1.2	Fußkraftsensoren	132
7.1.3	Orientierungssensor	133
7.1.4	GPS	136
7.1.5	Laserabstandssensor	137
7.2	Reaktive Laufmaschinensteuerung	138
7.3	Lokalisation	139
7.3.1	Odometrie	139
7.3.2	Orientierungskorrekturen	140
7.3.3	Positionskorrekturen	142
7.3.4	Zusammenfassung und Erkenntnisse	146
7.4	Umweltmodellierung für Laufmaschinen	147
7.4.1	Dimensionierung der Karten	147
7.4.2	Virtuelle Sensoren	148
7.4.3	Aufbau der Karten	149
7.4.4	Zusammenfassung und Erkenntnisse	154
7.5	Bahnplanung	154
7.5.1	Graben	155
7.5.2	Säule	157
7.5.3	Bahnplanung	158
7.5.3.1	Vollständig bekannte Karte	159
7.5.3.2	Unvollständige Karte	160
7.5.4	Zusammenfassung und Erkenntnisse	160
7.6	Adaption an das Gelände	163
7.6.1	Geländebewertung	163
7.6.2	Anpassung der Laufparameter	165
7.6.3	Fußaufsetzpunktplanung	165
7.6.4	Zusammenfassung und Erkenntnisse	166
7.7	Autonomes Laufen in natürlichem Gelände	166
7.7.1	Bahnverfolgung	166
7.7.2	Auftragsverwaltung	167
7.7.3	Waldszenario	167
7.8	Zusammenfassung und Ergebnisse	171
8	Zusammenfassung und Ausblick	175
A	Grundlagen zur Lokalisation	179
A.1	Koordinatensysteme zur Lokalisation	179
A.2	Repräsentation der Lage	180
A.3	Koordinatentransformationen	183
A.3.1	Innerhalb des ecef -System	183
A.3.2	Von ecef nach ltp	183

A.4	Mehrdimensionale Taylorentwicklung	184
B	Kurzeinführung in die Fuzzy-Logik	187
B.1	Fuzzy-Mengen und linguistische Terme	187
B.2	Operatoren auf Fuzzy-Mengen	189
B.3	Fuzzy-Implikation und Fuzzy-Inferenz	189
C	Realisierte Fuzzy-Bewerter	191
C.1	Odometrie – Bodenkontaktbewertung	191
C.2	Umweltmodellierung – Einfügen neuer Sensordaten	193
C.3	Adaption an die Geländebedingungen	193
	Symbolverzeichnis	203
	Abbildungsverzeichnis	209
	Tabellenverzeichnis	211
	Literaturverzeichnis	213

Kapitel 1

Einleitung

Die technische Überlegenheit des Rades verliert schlagartig an Bedeutung, wenn man sich überlegt, dass zum Rad eigentlich die Straße gehört.

(Nachtigall und Büchel, 2000)

Etwa die Hälfte der Landfläche der Erde ist für rad- oder kettengetriebene Fahrzeuge unzugänglich. Ein weiterer Teil der begehbaren Landfläche bleibt „konventionellen“ Fahrzeugen aus anderen Gründen wie beispielsweise dem Landschaftsschutz verwehrt. Rehe durchqueren solches Gelände hingegen problemlos, ohne die Landschaft zu stark zu beschädigen. In schwierigem Gelände eröffnet eine auf Beinen basierende Fortbewegung eine größere Mobilität. Einer der Gründe hierfür liegt in der Verwendung isolierter Fußaufsetzpunkte, um den Körper zu tragen und voranzutreiben, wohingegen Räder auf einen kontinuierlichen Pfad angewiesen sind. Eine Unterbrechung der ansonsten stetigen Oberfläche, beispielsweise durch einen Graben, stellt für radgetriebene Fahrzeuge in der Regel ein nicht zu überwindendes Hindernis dar. Robotern, die sich auf Beinen fortbewegen, bietet sich hingegen die Möglichkeit, den Graben durch geschickte Wahl der Fußaufsetzpunkte zu überqueren. Als weiteren Vorteil von Beinen führt Raibert (1985) die aktive Federung an, welche den Pfad des Körpers vom Pfad der Füße entkoppelt. Somit muss die Nutzlast im Gelände den Bodenunebenheiten nicht folgen, was die Geschwindigkeit und Effizienz in rauer Umgebung steigern kann.

Diesen Vorteilen stehen – hinsichtlich der technischen Umsetzung des Laufens – auch eine Reihe von Nachteilen gegenüber. Der Aufbau eines künstlichen Beines gestaltet sich im Vergleich zum Rad aufwendiger, was in der Regel mit einer Beschränkung der Nutzlast einhergeht. Weiterhin ist zur Realisierung von Beinbewegungen die Ansteuerung einer hohen Anzahl von Bewegungsfreiheitsgraden zu bewältigen.

Das für Laufmaschinen interessante Einsatzgebiet beginnt in Szenarien, wo dasjenige für Radfahrzeuge endet. Diese Arbeit behandelt die autonome Fortbewegung von Laufmaschinen in unstrukturiertem Gelände. Im Unterschied zu strukturierten Umgebungen lässt sich die Geometrie von unstrukturierten Gebieten nur mit großem Aufwand und erheblichen Unschärfen detailliert beschreiben, da kein vollständiges a-priori-Wissen über die zu betretende Oberfläche und die vorherrschenden Bodenbedingungen verfügbar ist. Vulkangebiete, Katastrophenzonen oder Waldgebiete sind Beispiele für unstrukturier-tes Gelände. Damit sich Laufmaschinen in derartigen Umgebungen sicher fortbewegen

können, wird es notwendig, die folgenden Schritte vorausschauend zu planen und gegebenenfalls das Laufverhalten anzupassen. Eine Vorausschau bedingt jedoch ein Modell der Umgebung, zu dessen Aufbau wiederum Wissen um die eigene Lage in Bezug auf die Umwelt erforderlich ist. Dies führt zum Begriff der Navigation.

1.1 Navigation

Der Begriff *Navigation* stammt aus dem lateinischen (*navigatio* „Schifffahrt“). Der Fremdwörterduden definiert *Navigation* folgendermaßen:

Bei Schiffen und Flugzeugen die Einhaltung des gewählten Kurses und die Standortbestimmung

(*Fremdwörterduden, 1982*)

Eine detailliertere Definition findet sich im Brockhaus-Lexikon:

Führung von Schiffen, Luft- oder Raumfahrzeugen
 astronom. Navigation (Standortbestimmung durch Höhenbeobachtung zweier Gestirne mittels Sextanten);
 Funknavigation;
 barometr. Navigation (Fliegen nach Druckflächen);
 terrestr. Navigation (optische Beobachtung landfester Ziele u.a.);
 die Satellitennavigation ermöglicht wesentlich genauere Ortsbestimmungen und damit Treibstoffeinsparungen

(*Brockhaus, 1984*)

In seiner ursprünglichen Bedeutung bezog sich der Begriff *Navigation* ausschließlich auf die Schifffahrt. Später wurde er auch in der Luft- und Raumfahrt verwendet. In den folgenden Jahren fand der Navigationsbegriff Einzug in weitere Bereiche mobiler Systeme wie das Verkehrswesen oder die Robotik. *Navigation* beschreibt dabei im wesentlichen vier Aspekte:

Standortbestimmung: Unter Einsatz unterschiedlicher Methoden wird die Position und ggf. die Orientierung eines mobilen Systems im Raum bestimmt.

Berechnung eines Kurses: Ein mobiles System soll sich zu einem vorgegebenen Zielort bewegen. Ausgehend von der aktuellen Position wird ein Kurs zum Erreichen dieses Zielortes ermittelt.

Erstellung einer Karte: Eine notwendige Voraussetzung für die Kursberechnung ist eine Karte, welche die befahrbaren und zur Verfügung stehenden Wege aufzeigt. Steht eine solche Karte von Beginn an nicht zur Verfügung, muss diese sukzessive aufgebaut werden.

Einhaltung eines Kurses/Führung: Um den Zielort schließlich erreichen zu können, muss das mobile System den geplanten Kurs verfolgen und ggf. auf Umwelteinflüsse reagieren.

Diese Arbeit befasst sich mit der Navigation von Laufmaschinen in natürlichem Gelände. Es wird dabei angenommen, dass für das zu durchquerende Gebiet noch keine Karte zur Verfügung steht – zumindest keine Karte, die den erforderlichen Detaillierungsgrad für ein sicheres Durchlaufen des Geländes aufweist. Hieraus ergibt sich zusätzlich die Notwendigkeit, während des Laufens eine Karte und somit ein Modell der Umwelt aus den zur Verfügung stehenden Sensordaten aufzubauen.

Überträgt man obige Ausführungen nun auf eine zu steuernde mobile Roboterplattform, lassen sich vier größere aufeinander aufbauende Teilprobleme unterscheiden:

Lokalisation: Exakte Bestimmung der aktuellen Position und Orientierung des Roboters in seiner Umwelt

Umweltmodellierung: Integration des durch Einsatz der Sensoren gesammelten Wissens über die reale Welt in einem Modell

Bahnplanung: Die Erstellung eines Plans zum Erreichen der gegebenen Zielvorgabe

Adaption: Eine an die örtlichen Gegebenheiten angepasste Ausführung des Plans

Jeder dieser vier Teilprobleme stellt ein in sich abgeschlossenes Themenfeld dar, das im Grunde genommen unabhängig von den jeweils anderen behandelt werden kann. Zwar zeigen sich methodische Abhängigkeiten sowie Gemeinsamkeiten bezüglich der Umweltrepräsentation, jedoch können diese auch bei getrennter Abhandlung der genannten Problemfelder an geeigneter Stelle zum Ausdruck gebracht werden.

1.2 Ziel der Arbeit

Betrachtet man die Veröffentlichungen im Bereich *Laufmaschinen*, so stellt man fest, dass sich der Großteil der Arbeiten dem mechatronischen Entwurf, dem Aufbau und der Basissteuerung dieser Art von Roboter widmen. Dies verwundert nicht, führt man sich die Komplexität von laufenden Maschinen vor Augen: Bis menschenähnliche Roboter realisierbar sind, die den Menschen wie beispielsweise im Film „I, Robot“ (Proyas, 2004) umfassend bei der Arbeit unterstützen können, wird noch einige Zeit intensive Forschung und Entwicklung notwendig sein. Das zunehmende wissenschaftliche Interesse am zweibeinigen Laufen humanoider Roboter ist ein wichtiger Schritt in diese Richtung, jedoch ist die Bewegung 2-beiniger Roboter zum aktuellen Zeitpunkt auf stark strukturierte Laborumgebungen begrenzt. Damit sich eine autonome Laufmaschine auch in unstrukturiertem Gelände fortbewegen kann, sind neben der grundsätzlichen Fähigkeit zu Laufen weitergehende Navigationsfähigkeiten erforderlich. Für Laufmaschinen in unstrukturiertem Gelände sind dem Autor umfassendere Ansätze nicht bekannt. Ziel dieser Arbeit ist es, einen Beitrag zur Schließung dieser Lücke zu leisten.

Der in dieser Arbeit betrachtete Ansatz soll Laufmaschinen in freiem Gelände in die Lage versetzen, einen Auftrag, der über einen Leitstand erteilt wird, selbstständig auszuführen. Solche Aufträge können z.B. die Form „Bewege dich ausgehend vom *Start* hin zum *Ziel*“ (Abbildung 1.1) haben. Zur erfolgreichen Durchführung dieser



Abbildung 1.1: Skizziert ist ein Aufgabenszenario, das eine Laufmaschine durch Einsatz von Navigationsalgorithmen autonom bewältigen soll. (a) Ausgehend von der aktuellen Position *Start* soll der Roboter einen begehbaren Weg zum *Ziel* finden. (b) Der Roboter muss einen geeigneten Weg autonom auswählen.

Aufgabe muss er fähig sein, sich in der aktuellen Umwelt zu orientieren und seine Lage ständig zu aktualisieren. Da von einer unbekanntem Umgebung ausgegangen wird, soll sukzessive eine Karte der Umwelt automatisch aufgebaut werden. In dieser Karte ist es dann möglich, einen Pfad hin zum Zielpunkt zu planen. Beim Ablaufen des Pfades soll der Roboter sein Laufverhalten abhängig von den Umgebungsbedingungen anpassen: wird das Gelände *schwieriger* soll der Roboter *vorsichtiger* laufen, wird es hingegen *einfacher* soll er sich durchaus *schneller* fortbewegen.

1.3 Beitrag der Arbeit

Um das oben genannte Ziel zu erreichen, werden die vier oben aufgeführten Teilaufgaben der Navigation, Lokalisation, Umweltmodellierung, Bahnplanung und Adaption, für eine Laufmaschine realisiert. Aus dem Bereich der mobilen Roboterplattformen existieren für viele dieser Problemfelder die unterschiedlichsten Lösungsansätze. Ein Teilaspekt dieser Arbeit ist die Untersuchung, ob und inwieweit bereits entwickelte Verfahren dazu beitragen können, eine modellbasierte, sensorgestützte Navigation für Laufmaschinen zu entwickeln. *Laufmaschinen in natürlichem Gelände* bringen in vielerlei Hinsicht spezielle Anforderungen und Randbedingungen mit sich. Um diesen Forderungen gerecht zu werden, werden sowohl einige bereits entwickelte Verfahren erweitert als auch neuartige Algorithmen und Vorgehensweisen vorgeschlagen.

Zum Nachweis der in dieser Arbeit vorgeschlagenen und konzeptionierten Fähigkeiten wird die am *Forschungszentrum Informatik (FZI)* entwickelte sechsbeinige Laufmaschine LAURON herangezogen. Die Sensorik der Laufmaschine wird für die Umsetzung der modellbasierten Navigation in natürlichem Gelände erweitert.

1.4 Aufbau der Arbeit

Im Folgenden wird der Aufbau der Arbeit erläutert. Tabelle 1.1 stellt den Aufbau der Arbeit zusammenfassend dar. Kapitel 2 definiert zunächst die allgemeinen Anforderungen und Randbedingungen einer modellbasierten, sensorgestützten Navigation für Laufmaschinen im Gelände. Anschließend werden die Navigation betreffende Aspekte von in der Literatur vorgestellten Laufmaschinensteuerungen und vergleichbaren radgetriebenen Systemen untersucht und hinsichtlich eines Einsatzes auf autonomen Laufmaschinen bewertet.

Die Lokalisationsproblematik ist Gegenstand des Kapitels 3. Anhand der Anforderungen und Randbedingungen der Lokalisation von Laufmaschinen werden verschiedene Verfahren bewertet. Danach werden die am besten geeigneten auf den Laufmaschineneinsatz abgestimmt und einem Fusionsprozess zugeführt. Das hierfür konzeptionierte Kalman-Filter wird hergeleitet und abschließend der Ablauf des Lokalisationsalgorithmus skizziert.

Kapitel 4 befasst sich mit der Umweltmodellierung. Zunächst werden die Anforderungen und Randbedingungen eines Umweltmodells für Laufmaschinen im Gelände erörtert. Auf deren Basis werden in der Literatur beschriebene Ansätze evaluiert und weiterverfolgt. Die Repräsentation der Umwelt in einer geometrischen dreidimensionalen Karte sowie der Übergang auf eine zweidimensionale Höhenkarte sind ein weiteres Thema des Kapitels.

Die Bahnplanung wird in Kapitel 5 behandelt. Sowohl allgemeine als auch laufmaschinenspezifische Planungsverfahren werden anhand zuvor formulierter Anforderungen und Randbedingungen evaluiert. Die Erkenntnisse hieraus bilden die Grundlage zu der Konzeption eines Bahnplaners für Laufmaschinen bestehend aus Laufmaschinenmodell, Bewertungsfunktion und Potenzialfeldverfahren.

Kapitel 6 befasst sich schließlich mit der Adaption. Nach Betrachtung vorhandener Geländebewertungen wird eine Beschreibung des Geländes mittels Kennzahlen hergeleitet. Es wird gezeigt, wie anhand dieser Information das Laufverhalten an die Geländebedingungen angepasst werden kann. Das Kapitel schließt mit der Betrachtung optimaler Fußaufsetzpunkte.

In Kapitel 7 wird zunächst die sechsbeinige Laufmaschine LAURON IV und deren dieser Arbeit zu Grunde liegende reaktive Laufmaschinensteuerung kurz präsentiert. Anschließend wird die Umsetzung der konzipierten Methoden und Verfahren aus den vorangehenden Kapiteln auf LAURON IV erläutert. Die Funktionstüchtigkeit der realisierten Komponenten wird anhand unterschiedlicher Experimente unter Beweis gestellt. Die Ergebnisse werden festgehalten und diskutiert.

Abschließend wird in Kapitel 8 eine kurze Zusammenfassung der wesentlichen Aspekte dieser Arbeit gegeben. Mit einem Ausblick auf zukünftige Arbeiten schließt letztlich diese Arbeit.

1	Einleitung
2	Navigation mobiler Roboter Modellbasierte Navigation – Anforderungen und Randbedingungen ▼ Laufmaschinen ▼ Radgetriebene und hybride Roboter
3	Lokalisation Anforderungen und Randbedingungen ▼ Relative und absolute Lokalisationsverfahren ▲ Odometrie, Orientierungssensor, GPS, Kalman-Filter
4	Umweltmodellierung Anforderungen und Randbedingungen ▼ Geometrische und topologische Karten, Belegtheitsgitter ▲ 3D-Urkarte, globales 2½D-Modell
5	Bahnplanung Anforderungen und Randbedingungen ▼ Allgemeine und laufmaschinenspezifische Planungsverfahren ▲ Laufmaschinenmodell, Bewertungsfunktion, Potenzialfeldverfahren
6	Adaption von Laufverhalten ▼ Bewertung von Gelände ▲ Beschreibung von Gelände, Laufparameterauswahl, Fußpunktselektion
7	Realisierung, Experimente und Ergebnisse ◆ LAURON IV, reaktive Laufmaschinensteuerung ◆ Aspekte der Realisierung, Experimente und Ergebnisse
8	Zusammenfassung und Ausblick
▼	Stand der Technik
▲	Konzept
◆	Umsetzung

Tabelle 1.1: Aufbau der Arbeit

Kapitel 2

Navigation mobiler Roboter

Where am I? Where am I going? How should I get there?

(Leonard und Durant-White, 1991a)

Mit diesen drei Fragen umschreiben Leonard und Durant-White (1991a) die generelle Problematik der Navigation mobiler Roboter:

1. Die Antwort auf die erste Frage „Wo bin ich?“ soll die Lokalisation geben: Wie kann ein Roboter auf Basis dessen, was er sieht und was er bisher weiß, berechnen, wo er sich in seiner Umgebung befindet?
2. Die Spezifikation eines Ziels ist die Antwort auf die zweite Frage: „Wo gehe ich hin?“ Zur Festlegung eines bestimmten Ziels kann eine Karte der Umgebung herangezogen werden.
3. Schließlich ist es Aufgabe der Planung, die Frage „Wie soll ich dorthin kommen?“ zu beantworten: Das *Wie* kann auf die explizite Angabe eines Weges hin zielen: das ist Aufgabe der Bahnplanung. Darüber hinaus kann auch die Art und Weise der Fortbewegung von Interesse sein: dies sollte die Adaption situationsbedingt entscheiden.

Im diesem Kapitel werden zunächst einige Laufmaschinen vorgestellt, die für den Einsatz in unstrukturiertem Gelände entwickelt wurden. Hierbei liegt das Hauptaugenmerk auf Systemen, deren Steuerungsansätze Aspekte der Navigation berücksichtigen. Es wird aufgezeigt, welche Methoden bezüglich der Lokalisation, Umweltmodellierung, Bahnplanung sowie Adaption zum Einsatz kommen. Weiterhin wird untersucht, inwieweit diese auf Laufmaschinen übertragbar sind. Schließlich werden geländefähige radgetriebene und hybride Robotersysteme betrachtet und deren Navigationsmethoden hinsichtlich deren Nutzung und Integration auf Laufmaschinensteuerungen diskutiert.

2.1 Allgemeine Anforderungen und Randbedingungen

Obwohl das gesuchte Navigationskonzept sowohl für beliebige Laufmaschinen als auch für beliebige Umgebungen geeignet sein soll, kann es nur schwer allen Ansprüchen gerecht werden. Ein Kleinroboter wie beispielsweise STIQUITO (McClain und Conrad,

2005) bewegt sich unter anderen Randbedingungen als eine große Laufmaschine wie zum Beispiel ROBOCLIMBER (Molfino et al., 2005). Die jeweils zur Verfügung stehenden Ressourcen sowie die unterschiedlichen Zielsetzungen der Roboter erfordern verschiedene Steuerungsansätze, so dass die Navigationsproblematik unterschiedlich gestaltet ist. Jedoch unterscheiden sich nicht alle Laufmaschinen derart wie die beiden oben genannten, so dass der in dieser Arbeit vorgestellte Navigationsansatz sich nicht nur auf die in Abschnitt 7.1 beschriebene Roboterplattform LAURON beschränkt.

Folgende grundlegend einschränkende Randbedingungen sind zu beachten:

Autonomie: Die Laufmaschine soll sich autonom fortbewegen können. Hierzu ist es notwendig, neben der Aktorik sowohl die gesamte einzusetzende Sensorik als auch die zu deren Verarbeitung genutzte Rechnerhardware auf dem Roboter mitzuführen. Um energieautark agieren zu können, muss die für die Dauer einer Mission benötigte Energie auf dem Roboter bereitgestellt werden.

Rechnereffizienz: Ein autonomer Roboter muss in der Lage sein, alle für die Bewegung und Planung notwendigen Berechnungen selbstständig durchzuführen. Die zur Verfügung stehende Rechnerkapazität auf einer Laufmaschine ist aus Energie-, Gewichts- sowie aus Platzgründen grundsätzlich beschränkt. Somit ist der sorgsame Umgang mit der Ressource „Rechenzeit“ bei der Entwicklung von Algorithmen erforderlich.

Einfache Sensorik: Neben der benötigten Rechenzeit beschränken vor allem Größe, Gewicht und Energieverbrauch die verwendbare Sensorik. Wenngleich der Einsatz hochauflösender Sensorik nicht grundsätzlich ausgeschlossen wird, kann deren Verfügbarkeit nicht vorausgesetzt werden.

Echtzeitfähigkeit: Um die erfolgreiche Durchführung einer Mission nicht zu gefährden, muss der Roboter in angemessener Zeit auf dynamische Ereignisse in der Umwelt reagieren und sich fortbewegen können.

2.2 Navigationsfähigkeiten von Laufmaschinen

In den letzten Jahren wurde weltweit an zahlreichen laufenden Robotern gearbeitet. Der Großteil dieser Projekte beschäftigt sich mit dem mechanischen Entwurf, dem Aufbau der Systeme und der Entwicklung von Steuerungsgrundlagen, die dem Roboter das Laufen überhaupt erst ermöglichen. Obgleich viele der mehrbeinigen Systeme (vergleiche beispielsweise TUM (Weidemann et al., 1994), GENGHIS (Celaya und Porta, 1998), TARRY (Frik et al., 1999) oder SILO4 (de Santos et al., 2003)) die Fähigkeit besitzen, Hindernisse zu übersteigen und unstrukturiertes Gelände zu durchqueren, ist es auf Grund der eingeschränkten Navigationsfähigkeit oft nicht möglich, sie zielgerichtet in eine bestimmte Richtung zu bewegen. Des Weiteren scheitert selbstständiges, vorausschauendes Planen meist an der fehlenden geeigneten Navigations- und Lokalisationssensorik, um entsprechende Umweltgegebenheiten wahrzunehmen. Umfassende

Ansätze für unstrukturiertes Gelände, welche über die Basissteuerung hinaus auch Navigationsaspekte berücksichtigen, sind selten.

Im Bereich der zweibeinigen Laufmaschinen werden vor allem Stereokamerasysteme genutzt, um Informationen über die unmittelbare Umgebung des Roboters zu gewinnen. Visuelle Sensorsysteme unterstützen die Suche nach ebenen Fußaufsetzflächen, ermöglichen es dem Roboter Treppen zu erklimmen und Hindernisse zu übersteigen oder gar durch Anpassung der Körperhaltung Engstellen zu passieren (vergleiche beispielsweise (Chestnutt et al., 2005; Lorch et al., 2002; Seara et al., 2002; Okada et al., 2005; Kumagai und Emura, 2000) oder (Kanehiro et al., 2005)). Jedoch sind nur wenige zweibeinigen Roboter in der Lage, sich autonom fortzubewegen: sie werden zwar über mitgeführte Batterien mit Energie versorgt aber oft mit Hilfe externer Rechneinheiten gesteuert. Darüber hinaus beschränkt sich der Einsatz 2-beiniger Laufmaschinen wie beispielsweise die *HONDA*-Roboter (Honda, 2004) oder HRP-2 (Kaneko et al., 2004) auf strukturierte, klar definierte Umgebungen. Dies schränkt die Übertragbarkeit derartiger Ansätze auf das in dieser Arbeit betrachtete Problemfeld des autonomen Laufens in unstrukturiertem Gelände ein.

Trotz der großen Vielfalt an Laufmaschinen¹ befassen sich nur wenige Forschungsvorhaben mit der Navigationsproblematik unter den speziellen Randbedingungen der Laufmaschinen in natürlichem Gelände. Dieser Abschnitt diskutiert im Folgenden einige Robotersysteme, deren Steuerungen Teilbereiche der Navigation berücksichtigen.

2.2.1 Adaptive Suspension Vehicle

Das *ADAPTIVE SUSPENSION VEHICLE* (ASV) ist eine zwischen 1981 und 1987 an der *Ohio State University* und der *Adaptive Machine Technologies* entwickelte experimentelle sechsbeinige Laufmaschine (Abbildung 2.1). Ein in einem 2600 kg schweren und 500×100×330 cm großen Roboter mitgeführter menschlicher Operator hat die Möglichkeit, zwischen verschiedenen Steuerungsmodi zu wählen. Von der direkten Ansteuerung einzelner Beine über die Ansteuerung der vorderen Führungsbeine bis zur Vorgabe der Zentralkörperbewegungsrichtung reicht das Bedienungsspektrum („shared control“) bei steigendem Autonomiegrad der Beinbewegungen (Bihari et al., 1989).

Auf der obersten Autonomiestufe wird durch den Einsatz eines 3D-Distanz-Laserscanners eine lokale Höhenkarte des Geländes generiert (Kau et al., 1989). Um die fortlaufenden Abstandsmessungen in der Karte zu vereinen, wird die Lage des ASV im Raum durch Einsatz eines inertialen Messsystems, bestehend aus Beschleunigungssensoren, Drehratengeber sowie Gyroskopen, ständig fortgeschrieben. Mit Hilfe dieser Karte wird ein vereinfachtes geometrisches Geländemodell aufgebaut, so dass die Bewegungsplanung die Vorgaben des Operators unter Einhaltung von Stabilitätskriterien simulieren kann (Lee und Orin, 1988). Darauf aufbauend werden die einzelnen Beintrajektorien generiert, wobei aus der Höhenkarte zusammenhängende ebene Bereiche als mögliche Fußaufsetzpunkte identifiziert werden.

¹Eine umfassende Auswahl an Beispielen kann unter www.walking-machines.org im Laufmaschinenkatalog abgerufen werden.



Abbildung 2.1: ADAPTIVE SUSPENSION VEHICLE – Ein experimenteller sechsbeiniger Roboter mit dem Menschen als „Fahrer“.

Hierdurch ist der Operator in der Lage, das ASV über unebenes Gelände mit kleineren Hindernissen, Hügeln und Senken zu steuern. Zum Überqueren von größeren Hindernissen muss ein spezieller Klettermodus aktiviert werden, in dem weitere vordefinierte Bewegungsmuster zur Verfügung stehen. Die Steuerung des ASV verfügt über keine höheren Bahnplanungsalgorithmen, da die Navigation größtenteils vom menschlichen Operator übernommen wird. Aus diesem Grund wird auf die Erstellung eines globalen Umweltmodells verzichtet, so dass die Erfassung der Roboterposition durch in der Regel driftende, relativ messende Systeme den Anforderungen genügt.

2.2.2 Ambler

Die sechs Beine von AMBLER, einer an der *Carnegie Mellon University (Pittsburgh)* entwickelte Laufmaschine, sind an zwei Seiten übereinander rotierend aufgehängt (Abbildung 2.2). In einer typischen Stemmphase hat die Laufmaschine die Ausmaße von etwa $350 \times 450 \times 500$ cm. Auf Grund des geplanten Einsatzgebietes, der Planetenerkundung, ist der Roboter auf weitestgehende Autonomie ausgelegt. AMBLER verfügt über einen 3D-Laserscanner, aus dessen Tiefenbildern eine geometrische Höhenkarte mit modellierter Unsicherheit des Geländes erstellt wird. Die Fusion mehrerer Tiefenbilder findet dabei ohne zusätzliche Positionsinformation statt; im Gegenteil liefert dieser Fusionsprozess sogar eine Schätzung des Roboterversatzes (Hebert et al., 1989). Eine weitere Methode zur Lokalisation der Laufmaschine wird in (Roston und Krotkov, 1992) vorgestellt: die Koppelnavigation auf Basis der Beinbewegungen (vergleiche hierzu auch Abschnitt 3.2.1.1).

Das Planungssystem verwendet die Höhenkarte, um einen hindernisfreien Pfad sowie geeignete Fußaufsetzpunkte zu finden (Wettergreen et al., 1990). Die Trajektorienplanung berechnet dabei zuerst alle Gebiete, auf denen AMBLER stabil stehen kann, um anschließend mögliche Pfade anhand der begehbaren Punkte zu generieren. Der beste



Abbildung 2.2: AMBLER – Eine Laufmaschine zur autonomen Erkundung von Planeten.

Pfad wird anschließend durch eine Bewertung der Pfade bezüglich Energieverbrauch, Zeitaufwand und Stabilität ausgewählt. Daraufhin werden Gangmuster geplant, die diesen Pfad effizient abarbeiten. Hierbei werden die jeweiligen Schrittparameter wie Schritttrichtung, -länge, -breite sowie die grundsätzliche Schrittfolge festgelegt. Durch das geplante Gangmuster sind die einzelnen Fußpunkte schon grob vorgegeben, so dass die Fußpunktplanung nur auf lokalen Bereichen arbeitet, um den exakten Aufsetzpunkt zu bestimmen. Dabei bewertet ein Künstliches Neuronales Netz mögliche Fußaufsetzpunkte auf Basis von Geländeeigenschaften wie Stabilität und Rutschigkeit (Simmons und Krotkov, 1991). Da diese jedoch nicht direkt messbar sind, finden statt dessen Indikatoren wie die Geländesteigung, Geländeunebenheit und Oberflächenverlauf Verwendung.

Der Steuerungsansatz von AMBLER umfasst alle Bereiche der Navigation: von Lokalisationsmethoden über die Modellierung der Umwelt bis hin zu Planungsalgorithmen werden alle Teilgebiete in einem System integriert. Auf Grund der riesigen Ausmaße des Robotersystems kann der Transport auf andere Planeten wie beispielsweise den Mars jedoch ausgeschlossen werden. Zudem ist der Aktionsradius des Roboters stark eingeschränkt, da er über Kabel mit einer Basisstation verbunden ist, um sowohl mit Steuerungssignalen der externen Rechereinheiten als auch mit der notwendigen Energie versorgt zu werden. Auf Grund der zur Verfügung stehenden Rechnerhardware waren die aufwändigen Berechnungen auch keineswegs in Echtzeit möglich, was die Bewegungsgeschwindigkeit von AMBLER einschränkte.

2.2.3 Dante II

DANTE II, ein achtbeiniger nach dem „sliding frame“-Prinzip² arbeitender Roboter, wurde an der *Carnegie Mellon University (Pittsburgh)* zur Erkundung von Vulkana-

²Die acht Beine sind an zwei gegenseitig verschieb- und verdrehbaren Rahmen befestigt. Während die Beine des einen Rahmens den Roboter unterstützen, können die des anderen neu platziert werden.



Abbildung 2.3: DANTE II – Ein Roboter zur Vulkanerkundung.

nen entwickelt (Bares und Wettergreen, 1999). Der Roboter mit den Abmessungen $370 \times 230 \times 370$ cm bei einem Gewicht von 770 kg wird durch ein Stahlkabel am Hang gehalten, über welches er auch mit Energie (2 kW) versorgt wird sowie diverse Telemetriedaten zur Verfügung stellt (Abbildung 2.3). Neben wissenschaftlicher Sensorik zur Vulkanerforschung ist DANTE II mit mehreren Überwachungskameras sowie einem 3D-Distanz-Laserscanner zur Erfassung der umliegenden Bodengeometrie ausgestattet.

Die Benutzerschnittstelle von DANTE II unterscheidet verschiedene Steuerungsmodi, die jeweils unterschiedliche Autonomiegrade des Roboters definieren (Tabelle 2.1). Im individuellen Aktuatorbetriebsmodus kann der Operator jedes Bein von DANTE II

Kontrollfunktion	Aktuator- modus	Rahmen- modus	Verhaltens- modus	Gang- modus	Pfad- modus
Regelung des Haltekabels					
Beinkoordination					
Körperhöhe und -lage halten					
Schritt anpassen					
Hindernisse übersteigen					
Gelände wahrnehmen					
Schritthöhe und -länge ermitteln					
Körperhöhe und -haltung ermitteln					
Fußpunkte korrigieren					
Orientierung ermitteln					
Pfadplanung, Hindernisvermeidung					

Tabelle 2.1: Die spezifizierten unterschiedlichen Steuerungsmodi und Autonomiegrade von DANTE II. Die stufenweise zuschaltbaren, automatischen Kontrollfunktionen erlauben einen flexiblen Einsatz (nach Bares und Wettergreen (1999)).

einzelnen bewegen, um dieses z.B. über ein größeres Hindernis zu bewegen. Im Rahmenmodus wird die Koordination der Beine automatisch durchgeführt. Bei der Erkundung des Mount-Spurr-Vulkans in Alaska kam hauptsächlich der Verhaltensmodus zum Einsatz: Unter Vorgabe von Verhaltensparametern wie Schrittweite, Körperhaltung und -höhe, passt sich DANTE II hierbei auf Basis der Beinkontakt- und Neigungssensoren an das Gelände an. Die Basisprimitiven für die Vorwärtsbewegung sind mit Hilfe einer Verhaltenssteuerung realisiert. Der autonome Gang-Betriebsmodus erfordert vom Operator die Eingabe einer zu verfolgenden Trajektorie, so dass der Roboter selbstständig aus der mit Hilfe eines 3D-Laserscanners generierten Höhenkarte die notwendigen Verhaltensparameter ermittelt. Die höchste definierte Autonomiestufe, der Pfad-Betriebsmodus, welcher Lokalisation, Bahnplanung sowie Hindernisvermeidung umfasst, wurde spezifiziert, aber nie umgesetzt.

2.2.4 JROB

Im Rahmen eines inter-universitären Forschungsprogramms an der *Tohoku University*, am *Tokyo Institute of Technology* und an der *University of Tokyo* wurden die vierbeinigen Laufmaschinen JROB-1 (Kagami et al., 1998) und JROB-2 (Konno et al., 2002) als Forschungsplattform zur Untersuchung der Kopplung von Perzeption und Aktion entwickelt (Abbildung 2.4). Hierzu wurde die Roboterbasis TITAN VIII weiterentwickelt und um einen Stereokamerakopf sowie leistungsfähige Rechnerhardware zur Bildverarbeitung erweitert. Das Gesamtsystem ist 40 cm lang, wiegt etwa 30 kg und wird ferngesteuert. Ein Schwerpunkt der Forschungsarbeiten liegt bei Methoden der Bildverarbeitung wie Farbsegmentierung, Berechnung des Optischen Flusses und des Stereotiefenbildes unter Berücksichtigung der durch das Laufen verursachten Vibrationen. Die aus den Kamerabildern gewonnenen Informationen werden anschließend benutzt, um beispielsweise Objekte oder ein vorgegebenes Zielgebiet zu verfolgen, aber

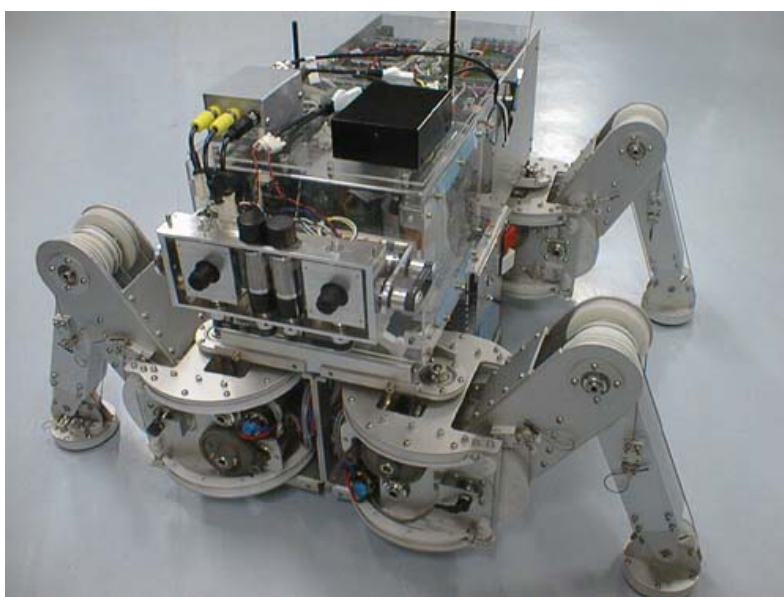


Abbildung 2.4: Die vierbeinige Forschungsplattform JROB-2.

auch um Hindernisse in der strukturierten Laborumgebung zu vermeiden. Igarashi und Kakikura (2004) stellen eine auf Potenzialfeldern basierende Planungsmethode vor, mit Hilfe derer ein Pfad zu einem vorgegebenen Zielpunkt geplant wird. Dabei wird berücksichtigt, dass Hindernisse je nach Höhe und Breite durch Übersteigen, aber auch durch Ducken passiert werden können. Die Eigenschaften des stark an Bai et al. (2002) angelehnten Ansatzes werden in Abschnitt 5.4.2 ausführlich erläutert.

2.2.5 RHex

Der dynamische sechsbeinige Roboter RHex wurde in einem Gemeinschaftsprojekt der *University of Michigan*, *McGill University*, *The University of California at Berkeley* und *Carnegie Mellon University* entwickelt. Der energieautarke Roboter RHex hat eine Größe von $47 \times 40 \times 24$ cm und wiegt durch einfaches mechanisches Design nur 8,5 kg. Durch unterschiedliche Ansteuerung der sechs unabhängig um die Hüfte rotierenden Beine kann der Roboter laufen, rennen sowie springen und klettern. Auch stark zerklüftetes und unebenes Gelände kann von RHex bei relativ hoher Geschwindigkeit von etwa 50 cm/s durchquert werden (Abbildung 2.5). Durch einfache Sensorrückkopplungen werden unterschiedliche Open-Loop-Steuerungsstrategien angepasst, um wechselnden Umweltbegebenheiten gerecht zu werden (Komsuoglu et al., 2001). Nach Skaff et al. (2003) konzentrieren sich die Navigationsansätze auf das Geradeauslaufen durch Einsatz eines Inertialsystems und die Verfolgung von sich klar vom Untergrund abhebenden Linien mit Hilfe einer Kamera. Durch Messen der Verformung der nachgiebigen Beine gelingt Lin et al. (2005) eine Schätzung der Roboterlage auf Basis der Odometrie. Das sehr spezielle Antriebskonzept dieses Robotersystems erlaubt es nicht, die speziell auf RHex zugeschnittenen Navigationsansätze auf Laufmaschinen im Allgemeinen zu übertragen.



Abbildung 2.5: Die sechsbeinige Laufmaschine RHex mit rotierenden Beinen.



Abbildung 2.6: Die achtbeinige biologisch motivierte Laufmaschine SCORPION.

2.2.6 Scorpion

Der Fokus des SCORPION-Gemeinschaftsprojektes der *Universität Bremen* und der *NASA* liegt auf der Entwicklung eines biologisch motivierten, achtbeinigen, geländefähigen Laufroboters (Abbildung 2.6). SCORPION hat eine Größe von $65 \times 40 \times (15-35)$ cm bei einem Gesamtgewicht von 11,5 kg. Zur Realisierung der Steuerung steht ein Netzwerk aus Motorola-MPC555-Mikrocontrollern zur Verfügung. Die Laufbewegungen des Roboters werden mit Hilfe so genannter zentraler Mustergeneratoren erzeugt, die rhythmische Bewegungsmuster sowie Haltungskontrollprimitive zur Verfügung stellen. Parallel hierzu sorgt ein auf sensomotorischen Rückkopplungen basierendes Reflexsystem für die Reaktion auf umweltbedingte Störungen wie Hindernisse (Spenneberger und Kirchner, 2002). Höhere Verhaltensweisen des Roboters nutzen Ultraschallsensoren zur Hindernisvermeidung sowie kompassbasierte Sensoren zur Einhaltung der vom Benutzer vorgegebenen Bewegungsrichtung. Zur Stabilisierung des Roboters wird weiterhin die Laufgeschwindigkeit an das Gelände angepasst, indem die Häufigkeit der auftretenden Hindernisreflexe sowie der Lagesensor ausgewertet werden. Robustheit und Fehlertoleranz von SCORPION stehen im Vordergrund der Entwicklung (Spenneberger et al., 2004), weitergehende Navigationsansätze sind nicht bekannt.

2.3 Radgetriebene und hybride Roboter

Dieser Abschnitt stellt einige mobile Plattformen für den Einsatz im Gelände vor, die entweder mit Laufmaschinen eng verwandt sind oder auf Grund ihrer vorgegebenen Randbedingungen Ähnlichkeiten mit den in dieser Arbeit betrachteten Laufmaschinen aufweisen.



Abbildung 2.7: Der an der *Helsinki University of Technology* entwickelte mobile zentaurenartige Serviceroboter WORKPARTNER mit hybridem Antriebssystem.

2.3.1 WorkPartner

WORKPARTNER ist ein am *Intelligent Machine and Special Robotics Institute* der *Helsinki University of Technology* entwickelter mobiler zentaurenartiger Roboter mit hybridem Antriebssystem (Ylönen und Halme, 2002). Der hauptsächlich für den urbanen Einsatz bestimmte Roboter besteht aus einer Plattform mit vier Beinen, an deren Ende jeweils ein Radantrieb angebracht ist, sowie einem zweiarmigen Torso zur komplexen Manipulation (Abbildung 2.7). Das hybride Antriebssystem kombiniert die Vorteile einer flexiblen und anpassungsfähigen Fortbewegung in schwierigem Gelände mit einem großen Geschwindigkeitsbereich von bis zu 7 km/h auf ebenem Untergrund. WORKPARTNER hat ein Gesamtgewicht von 250 kg bei einer Nutzlast von bis zu 40 kg. Dabei versorgt ein 3 kW Verbrennungsmotor den Roboter mit der notwendigen Energie.

An das Robotersystem werden anspruchsvolle Navigationsanforderungen gestellt: die weitläufige Navigation zur Überbrückung der Distanzen zwischen den Einsatzorten erfordert Genauigkeiten von etwa einem halben Meter, wohingegen die Navigation am Einsatzort mit einer Präzision von einigen Zentimetern durchgeführt werden muss, um die Manipulation von Objekten zu ermöglichen (Selkänaho et al., 2001). Die Lokalisationsbasis auf flachem ebenen Grund bildet die Odometrie der Antriebsräder sowie ein inertiales Messsystem. In schwierigem Gelände oder beim Laufen ist die Odometrie jedoch nicht vertrauenswürdig, so dass hier ausschließlich auf ein komplementäres System aus GPS-Empfänger und 2D-Laserscanner zurückgegriffen wird. In offenem Gelände stehen robuste GPS-Daten zur Verfügung, wohingegen nahe an Gebäuden oder Bäumen, wo der GPS-Empfang abgeschattet ist, die Bewegung des Roboters mit Hilfe von natürlichen Landmarken in überlappenden Laserscans errechnet wird.



Abbildung 2.8: Im *Jet Propulsion Laboratory* eingesetzte Testplattform ROCKY 7.

2.3.2 Mars Rover

Die Forschung im Bereich von Roboterplattformen zur Planetenerkundung gewann in den letzten Jahren durch die erfolgreich durchgeführten unbemannten Marsmissionen „Pathfinder“ (1997, mobiler Roboter SOJOURNER mit 9 kg Gewicht) sowie „Mars Surveyor Rover Mission“ (2004, mobile Roboter SPIRIT und OPPORTUNITY mit je 185 kg Gewicht und 100 m Reichweite pro Tag) enorm an Bedeutung. Die entfernungsbedingten Kommunikationstotzeiten einer Mission fernab der Erde erfordern hierbei weitestgehend autonome Roboter, die in einer unbekanntem unstrukturierten Umgebung navigieren, um das Hauptziel der Mission, wissenschaftliche Untersuchungen des Planeten, zu unterstützen. Durch den Einsatz von bewährter und raumfahrttauglicher Rechnerhardware ist die Rechenleistung auf den Rovern eingeschränkt. Darüber hinaus muss ein Rover seine Fähigkeiten bei minimalem Energieverbrauch entfalten können. Diese Anforderungen und Randbedingungen finden sich auch bei Laufmaschinen wieder (vergleiche hierzu Abschnitt 2.1).

Die mobile sechsrädrige Roboterplattform ROCKY 7 ist etwa $60 \times 40 \times 35$ cm groß und wiegt 15,5 kg (Abbildung 2.8). Als Rechereinheit steht ein cPCI-basiertes 750MHz PowerPC-System zur Verfügung. Die Roboterplattform verfügt über drei Beschleunigungssensoren zur Detektion der Neigung bezüglich des Untergrundes und einem Drehratensensor zur Orientierungsbestimmung, welcher auf Grund relativer Messungen einer Drift unterworfen ist. Eine absolute Orientierungsbestimmung mit Hilfe von Magnetfeldsensoren ist auf dem Mars wegen des schwachen magnetischen Feldes nicht möglich, weshalb die Orientierung durch Beobachtung des Sonnenstandes ermittelt wird (Volpe, 1999). Auf deren Basis wird die odometrische Positionsschätzung des Rovers berechnet. ROCKY 8, eine etwa doppelt so große Nachfolgeplattform, verfügt über drei Beschleunigungssensoren und drei Drehratensensoren.

An einem teleskopartigen Manipulator mit drei Bewegungsfreiheitsgraden ist ein Stereokamerasystem befestigt, das Panoramatiefenbilder der Umgebung liefern kann.

Aus diesen wählt der Operator täglich eine Reihe von Wegpunkten aus, die der Rover anfahren soll. Hierzu trägt der Rover die aus den Stereobildern extrahierten Hindernisse in eine einfache 2D-Karte ein, durch welche anschließend mit Hilfe einfacher Verhaltensregeln die nächste Bewegungsprimitive {0,5 rad auf der Stelle drehen, 0,25 m vorwärts, nach rechts oder nach links fahren} ausgewählt wird. Zwei weitere am Rumpf des Rovers angebrachte Schwarzweiß-Kameras werden zur lokalen Hindernisvermeidung verwendet.

Zur Unterstützung der Lokalisation werden die oben beschriebenen Methoden unter Einsatz eines Kalman-Filters zu einer Zustandsschätzung erweitert, so dass visuelle Lokalisationsmethoden wie Abgleich der Höhenkarten aus Stereobildern oder Berechnung der Stereo-Eigenbewegung die Lageschätzung verbessern können (Cheng et al., 2005). Derartige Berechnungen finden jedoch nicht in Echtzeit statt; sie benötigen vor Ort etwa 2–3 Minuten. Ein auf Fuzzy-Logik basierender Ansatz zur Anpassung der Roverbewegung an die Geländeeigenschaften wird in (Seraji und Howard, 2002) vorgestellt. Hierbei wird das lokale, regionale und globale Gelände jeweils auf seine Begehrbarkeit hin bewertet (vergleiche hierzu auch Abschnitt 6.1.2).

In zukünftigen Marsmissionen sollen Rover auf der Marsoberfläche innerhalb nur weniger Tage Gesteinsproben sammeln, die anschließend wieder zurück auf die Erde gebracht werden (Volpe et al., 2000; Volpe, 2005). Somit kommt der täglichen Reichweite eines Rovers eine enorme Bedeutung zu, so dass die oben vorgestellte Missionssteuerung nicht mehr angewandt werden kann. Die Anforderungen an die Rover-Steuerung steigen somit in vielen Bereichen:

- Das Umplanen der täglichen Aktivitäten in Abhängigkeit von nicht vorhersehbaren Gegebenheiten wie Positionsfehler, Geländeart, verbleibende Energie und Zeit muss vorgesehen werden.
- Eine autonome Bahnplanung wird notwendig, um innerhalb eines Tages über das Sichtfeld von 20 Metern hinaus agieren zu können.
- Der Rover muss über eine exakte und robuste Positionsbestimmung verfügen, um eine fehlerfreie autonome Fahrt insbesondere wieder zurück zur Basisstation sicher zu stellen.

2.4 Zusammenfassung und Schlussfolgerungen

Die ersten Laufmaschinen, die über Navigationseigenschaften verfügten, wie das ASV, AMBLER oder später auch DANTE II, waren alle sehr groß. Sie führten ihre Energieversorgung in Form eines Motors mit sich oder wurden von außen über ein Kabel mit Energie versorgt. Somit war es möglich, diese Roboter beispielsweise mit einem 3D-Laserscanner auszustatten, welcher eine lokale Höhenkarte der Umgebung bereitstellte und somit eine lokale vorausschauende Adaption an das Gelände unterstützte. Derart schwere und große Sensoren bleiben autonomen Laufmaschinen auch heutzutage meist verwehrt. Um die Aufgaben der Bahnplanung und Adaption auf Basis eines Umweltmodells durchführen zu können, muss eine autonome Laufmaschine deshalb in die Lage

versetzt werden, ein Modell der Umwelt mit leichter, energieeffizienter Sensorik aufzubauen.

Aktuelle Laufmaschinen wie JROB oder RH Ξ X nutzen wesentlich leichtere, dafür aber rechenintensive Kamerasysteme, um Informationen über die Umgebung zu gewinnen. Diese werden hauptsächlich zur Hindernisvermeidung sowie zur Planung des Pfades verwendet. Eine Adaption des grundlegenden Laufverhaltens an die nähere Umgebung findet nicht statt. Insbesondere werden keine dedizierten Fußaufsetzpunkte ermittelt, um ein sicheres Durchqueren von unstrukturiertem Gelände zu unterstützen. Die Roboterplattform SCORPION verfügt nur über rudimentäre Sensorik zur Erfassung der Umwelt, so dass auch hier eine vorausschauende Planung und Adaption nicht möglich ist.

Der Einsatz eines 2D-Laserscanners mit großer Reichweite zur Erfassung entfernter Landmarken wie beim hybriden mobilen Roboter WORKPARTNER ist mit autonomen Laufmaschinen noch nicht realisierbar, wenngleich leichte Distanzsensoren mit kurzer Reichweite verfügbar sind und zur Umwelterfassung von Laufmaschinen beitragen können (vergleiche Abschnitt 7.1.5). Die in zahlreichen Entwicklungen eingesetzte Stereokamerasysteme setzen entweder Spezialhardware (JROB-2) und/oder energieintensive Prozessoren (WORKPARTNER, diverse zweibeinige Laufmaschinen) oder aber lange Rechenzeiten (MARS ROVER) voraus.

Zusammenfassend lässt sich feststellen, dass für Laufmaschinen im Gelände kein umfassender Navigationsansatz verfügbar ist. Entweder ist eine zielgerichtete Navigation im Gelände überhaupt nicht möglich oder es werden die Anforderungen und Randbedingungen bezüglich Autonomie, Rechereffizienz, einfacher Sensorik oder Echtzeitfähigkeit verletzt. Die Intention dieser Arbeit ist es, einen Beitrag zur Schließung dieser Lücke zu leisten.

Kapitel 3

Lokalisation

Lokalisation (lat. locus, Ort) bedeutet ursprünglich, den Ort eines Geschehens oder Objektes festzustellen oder einzugrenzen. Der Begriff wird beispielsweise in der Akustik, in der Medizin, in der Technik und in der Informatik verwendet.

(Wikipedia, 2004)

Im Zusammenhang mit mobilen Systemen ist das Ziel der Lokalisation die möglichst exakte Bestimmung der aktuellen Position und Orientierung in der Umwelt. Alle weiteren Teilbereiche der Navigation nutzen das Wissen über die Lage¹ des Roboters im Raum: Alle über die Sensoren erfassten Umgebungseigenschaften, die in einer Karte gespeichert werden sollen, sind mit entsprechenden Lageinformationen verknüpft. Sinkt die Lokalisationsgenauigkeit, so reduziert sich die Qualität der Umweltbeschreibung. Durch geeignete Strategien in der Umweltmodellierung können Unzulänglichkeiten der Lagebestimmung zwar grundsätzlich abgemildert werden, jedoch erhöht sich der Aufwand, um den Qualitätsverlust in Grenzen zu halten. Die aktuelle Position und Orientierung stellen für die Bahnplanung die Ausgangsinformation dar. Je größer der Lokalisationsfehler zum Planungszeitpunkt, desto unbrauchbarer ist der darauf basierende Plan. Somit kommt dem Wissen über die Lage im Raum eine besondere Bedeutung zu. Im Folgenden wird beschrieben, welche Lokalisationsmethoden für Laufmaschinen in Betracht kommen und wie diese Methoden zu einer einheitlichen Lagebeschreibung kombiniert werden können. Im Rahmen von Studien- und Diplomarbeiten (Wiedemann, 2003; Zacharias, 2004; Haug und Rühl, 2005) wurden hierzu wertvolle Beiträge geleistet.

3.1 Anforderungen und Randbedingungen

Entsprechend der Zielsetzung dieser Arbeit soll die Laufmaschine in der Lage sein, sich autonom zu einem vorgegebenen Zielpunkt zu bewegen. Die hierdurch gestellten Anforderungen und Randbedingungen werden im Folgenden erörtert.

¹In dieser Arbeit soll im Weiteren unter dem Begriff *Lage* sowohl die Position als auch die Orientierung eines Körpers im Raum verstanden werden.

Begrenzter Lagefehler: Wird es von der Anwendung nicht ausdrücklich gefordert, kann zwar auf eine hohe absolute Lokalisationsgenauigkeit verzichtet werden. Eine definierte Zielkonfiguration kann jedoch nur dann zuverlässig erreicht werden, wenn der Lagefehler grundsätzlich begrenzt bleibt.

Unerwünschte Positionssprünge: Die Navigationsproblematik im Gelände erfordert zumindest den Aufbau einer lokalen Umgebungskarte. Positionssprünge während des Aufbaus der lokalen Karte stellen in der Regel enorme Störeinflüsse dar, die sich in der Kartenqualität negativ bemerkbar machen.

3.1.1 Besondere Anforderungen in unstrukturiertem Gelände

Die Zielstellung dieser Arbeit sieht die Navigation im Gelände vor; in natürlichem Gelände muss aber mit unstrukturierter Beschaffenheit gerechnet werden (vergleiche Kapitel 1). Hierdurch ergeben sich bei der Wahl von Lokalisationsalgorithmen zusätzliche Anforderungen und Einschränkungen:

Unabhängigkeit von a-priori-Karten: Da detaillierte Karten der Umgebung vorab selten verfügbar sind, jedoch die Planung ein Modell der Umwelt erfordert, muss neben der Lagebestimmung gleichzeitig ein Umweltmodell aufgebaut werden. Auf Karten basierende Lokalisationsverfahren sind somit eng an die Kartenerstellung gebunden.

Unabhängigkeit von strukturierten Umweltmerkmalen: Strukturierte Merkmale sind, wie der Name schon andeutet, in unstrukturiertem Gelände kaum zu finden. Lokalisationsverfahren, die sich derartiger Merkmale, wie z.B. Kanten, bedienen, sind nicht einsetzbar.

Robustheit gegen wechselnde Umgebungsbedingungen: Durch Sensorsysteme werden physikalische Eigenschaften der Umgebung gemessen. Die Algorithmen zur Auswertung der Sensorik müssen der Tatsache Rechnung tragen, dass sich natürliche Umgebungen ständig verändern (man denke z.B. an eine schwankende Temperatur oder wechselnde Lichtverhältnisse).

Robustheit gegen schwierige Bodenbedingungen: Unstrukturiertes Gelände kann Bodenunebenheiten oder weichen Untergrund mit sich bringen. Infolgedessen tendieren mobile Plattformen zum Rutschen oder Einsinken; die Roboterbewegungen werden unruhiger. Dies wiederum erfordert zusätzliche Anstrengungen bei der Sensordatenauswertung: Räder drehen durch, Kamerabilder verwackeln, usw.

3.1.2 Zusätzliche Anforderungen bei Laufmaschinen

Laufmaschinen zeichnen sich durch ihre hohe Flexibilität und Anpassungsfähigkeit aus; aber gerade auch diese Flexibilität erschwert u.a. die Lagebestimmung in der Umwelt. Sollen Laufmaschinen im Gelände lokalisiert werden, so ergeben sich zusätzlich zu den im letzten Abschnitt aufgezeigten Schwierigkeiten weitere zu berücksichtigende Randbedingungen:

Omnidirektionale Bewegung mit 6 Freiheitsgraden: Im dreidimensionalen Raum umfasst eine Lagebeschreibung sowohl die dreidimensionale Position mit den Komponenten x , y und z , als auch die dreidimensionale Orientierung mit den Komponenten α , β und γ (vergleiche Anhang A). Bei radgetriebenen Fahrzeugen wird insbesondere in flachem Gelände oft vereinfachend von $z = 0$, $\alpha = 0$, $\beta = 0$ ausgegangen. Derartige Annahmen verbieten sich bei Laufmaschinen, da Laufmaschinen den Zentralkörper beispielsweise schon im Stillstand in allen 6 Dimensionen bewegen können.

Punktuellem Bodenkontakt: Laufmaschinen nutzen Beine für die Fortbewegung. Die Beine unterliegen einem Bewegungszyklus, der in eine Stemmphase und in eine Schwingphase unterteilt wird. Die Stemmbeine stehen fest auf dem Boden, tragen das Gewicht des Roboters und schieben den Zentralkörper voran. Die Schwingbeine bewegen sich belastungsfrei durch die Luft zum nächsten Auftrittspunkt. Diese beiden Phasen wechseln sich zyklisch ab, um den Roboter voranzutreiben. Somit verfügt eine Laufmaschine, im Gegensatz zu radgetriebenen Fahrzeugen, über keinen durchgängigen Bodenkontakt.

Viele Bewegungsfreiheitsgrade: Ein Fahrzeug verfügt häufig nur über zwei Bewegungsfreiheitsgrade, eine Laufmaschine wie z.B. LAURON besitzt hingegen 18 (vergleiche Abschnitt 7.1). Bedingt durch die hohe Anzahl an Bewegungsfreiheitsgraden werden die Roboterbewegungen im Allgemeinen unruhiger und Vibrationen nehmen zu. Dies kann die im letzten Abschnitt skizzierten Probleme bei der Sensordatenauswertung unter schwierigen Bodenbedingungen noch verstärken.

Langsame Bewegung: Die Fortbewegungsgeschwindigkeit von Laufmaschinen liegt meistens weit unter denen von mobilen Plattformen. Auf den ersten Blick scheint diese Tatsache für die Lokalisation von Vorteil zu sein. Jedoch kann die langsame Geschwindigkeit beispielsweise die Messbereichsgrenzen von Sensoren unterschreiten.

3.2 Relative und absolute Verfahren

Die für die Lokalisation mobiler Systeme zur Verfügung stehenden Verfahren können grundsätzlich anhand der daraus gewonnenen Informationen in *relative* und *absolute* Verfahren unterteilt werden: Unter relativen Informationen versteht man Angaben, die sich auf eine definierte Ausgangskonfiguration beziehen. Informationen über die Lage ergeben sich aus der Ausgangslage und den zwischenzeitlich gemessenen Konfigurationsänderungen. Fehler in der Ausgangslage können durch die Messung nicht erfasst werden, so dass der Lagefehler grundsätzlich unbegrenzt wachsen kann. Absolute Verfahren sind hingegen unabhängig von der jeweiligen Ausgangskonfiguration. Die Lageinformation lässt sich direkt aus den Sensormessungen ableiten, wodurch der Lagefehler von der Messung und dem eingesetzten Verfahren bestimmt wird und sich somit in der Regel begrenzen lässt. Absolute Verfahren benötigen jedoch externe Referenzen oder

Umweltmerkmale, die unter Umständen nicht direkt verfügbar sind. Darüber hinaus erfordern diese oftmals einen hohen algorithmischen Aufwand, was die Häufigkeit der Lageaktualisierung einschränkt. Im Folgenden werden einige Lokalisationsverfahren vorgestellt und bezüglich der im letzten Abschnitt formulierten Anforderungen bewertet. Für einen umfassenderen Einblick in Sensoren und Standardverfahren zur Lokalisation mobiler (radgetriebener) Roboter sei z.B. auf Borenstein et al. (1996) verwiesen.

3.2.1 Koppelnavigation

Koppelnavigation (engl. Dead Reckoning) ist ein mathematisches Verfahren, um die aktuelle Position eines Objektes, ausgehend von der vorherigen Position, dem bekannten Kurs sowie dem zurückgelegten Weg, zu berechnen. Alle relativen Verfahren arbeiten nach dem Prinzip der Koppelnavigation. Die bekanntesten Vertreter dieses Verfahrens sind die Odometrie und die Inertialen Navigationssysteme.

3.2.1.1 Odometrie

Odometrie leitet sich aus dem Griechischen ab: *hodés*, der „Weg“ und *métron*, das „Maß“. Als *Odometrie* wird oftmals die einfachste Realisierung der Koppelnavigation bezeichnet. Der Begriff impliziert, dass der zurückgelegte Weg direkt durch einen mitgeführten Wegmesser bestimmt wird. Bei radgetriebenen Fahrzeugen geschieht dies in der Regel mit Hilfe von Encodern, welche die Bewegung von Lenk- bzw. Antriebsrad messen. Abhängig von der Fahrzeugkinematik, der Anordnung der Antriebs-, Lenk- und Messräder, können die zurückgelegte Distanz sowie der überstrichene Winkel mit Hilfe einfacher geometrischer Betrachtungen berechnet werden (Abbildung 3.1). Die bei der Berechnung auftretenden Fehler werden in zwei Kategorien unterteilt: Systematische Fehler sind bedingt durch nicht exakt bestimmte geometrische Parameter wie Raddurchmesser und Radabstände. Zur Klasse der nicht-systematischen Fehler zählen beispielsweise das Auftreten von Bodenunebenheiten und der Schlupf der Räder.

Viele Algorithmen haben zum Ziel, die durch Fehler bedingte Positionsunschärfe bzw. -ungewissheit zu schätzen. In Abbildung 3.1 wird diese Unschärfe mit Hilfe von so genannten *Fehlerellipsen* dargestellt: Ohne Korrektur von außen wächst die Ungewissheit über die „wirkliche“ Position des Roboters schrittweise an. Zur Reduktion der Fehler stellen z.B. Borenstein und Feng (1994) eine Kalibrierungsmethode vor, um systematische Odometriefehler experimentell zu ermitteln und diese anschließend zu korrigieren. Der Lokalisationsfehler kann jedoch erst in Kombination mit absoluten Verfahren beschränkt werden. In Verbindung mit Absolutkorrekturen setzen Martinelli und Siegwart (2003) mehrere Kalman-Filter ein, um sowohl systematische als auch nicht-systematische Fehlerkomponenten während der Fahrt zu schätzen, damit die Odometrie schließlich zuverlässiger wird.

Im Gelände wächst der nicht-systematische Fehler jedoch sehr schnell an, so dass oftmals ganz auf die Odometrie verzichtet wird. Ansätze zur Schätzung der Rad-Boden-Interaktion wie sie z.B. Andrade et al. (1998) vorstellen sind oft komplex und nicht für alle Geländearten geeignet. Spätestens in zerklüftetem Gelände muss auch die ver-

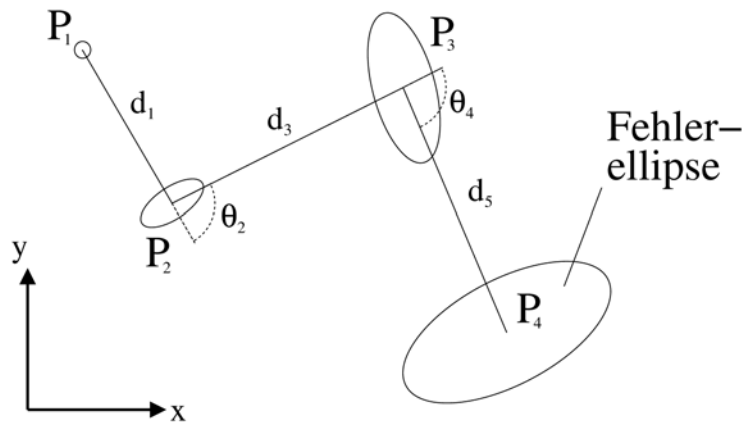


Abbildung 3.1: *Koppelnavigation* ermöglicht die Lageschätzung einer in Punkt P_1 startenden mobilen Plattform durch Beobachtung der zurückgelegten Wege d_1 , d_3 und d_5 sowie der überstrichenen Winkel θ_2 und θ_4 . Je weiter sich die Plattform vom Startpunkt entfernt, desto größer werden die in der Grafik durch Fehlerellipsen angedeuteten Unsicherheiten. Die spezielle Ausprägung der Fehlerellipsen hängt sowohl von der Roboterkonfiguration als auch von den berücksichtigten Unsicherheiten ab.

einfachende Annahme einer zweidimensionalen Roboterbewegung aufgegeben werden (Lamon und Siegwart, 2003).

Der überwiegende Teil der mobilen radgetriebenen Robotersysteme nutzt die Odometrie als Basissystem ihrer Lokalisation. Die Odometrie liefert kurzzeitig eine akzeptable Genauigkeit, ist nicht teuer und erlaubt sehr hohe Abtastraten. Da es sich bei der Odometrie jedoch um ein relatives Verfahren handelt und Fehler der Vergangenheit nicht korrigiert werden können, ist der auftretende Lokalisationsfehler nicht begrenzt. Warum die Odometrie trotzdem in fast allen mobilen Robotern zum Einsatz kommt, begründen Borenstein et al. (1996) folgendermaßen:

- Odometriedaten können mit absoluten Lagemessungen fusioniert werden, um eine bessere und vertrauensvollere Lageschätzung zu erhalten.
- Zur absoluten Lagebestimmung ist oftmals hoher algorithmischer Aufwand notwendig wie z.B. die Extraktion von Umgebungsmerkmalen. In der Zeit zwischen zwei absoluten Positionsangaben kann deshalb die Odometrie eingesetzt werden, um ständig aktuelle Lageinformationen zur Verfügung zu stellen.
- Viele absolute karten- oder landmarkenbasierte Verfahren erlauben unter Angabe einer fast korrekten Ausgangslage eine stark reduzierte Rechenzeit und eine höhere Korrektheit.
- In gewissen Situationen können absolute Verfahren auf Grund fehlender externer Referenzen versagen, z.B. wenn die Umgebung keine verwertbaren Landmarken aufweist. In diesen Fällen bietet die Odometrie die einzig verfügbare Lageschätzung.

Laufmaschinen besitzen keine Räder und verfügen nur über punktuellen Bodenkontakt. Odometriedaten mit deren vorteilhaften Eigenschaften bleiben aber auch Laufmaschinen nicht verwehrt: Roston und Krotkov (1992) präsentieren einen Ansatz, um die Odometrie von statisch stabil laufenden Maschinen zu berechnen. Aus den Stellungsänderungen der Stemmbeine werden mit Hilfe der Methode der kleinsten Fehlerquadrate auf Lageänderungen des Zentralkörpers geschlossen. Zur Reduktion nicht-systematischer Fehler werden als rutschend erkannte Beine bei der Berechnung ignoriert, was bei der Laufmaschine AMBLER (vergleiche Abschnitt 2.2.2) zu einer Ergebnisverbesserung führte. Der bei radgetriebenen Systemen durch Bodenunebenheiten hervorgerufene nicht-systematische Fehler spielt bei dieser Art der Berechnung keine Rolle, da grundsätzlich 3D-Lageänderungen betrachtet werden. Dieser Ansatz scheint vielversprechend und soll deshalb in dieser Arbeit als erste systemimmanente² Positionsschätzung weiter verfolgt werden (vergleiche Abschnitt 3.4). Ob der Aufwand einer Betrachtung von systematischen Fehlern wie sie z.B. Mae et al. (2001) vorstellen für die Zielsetzung dieser Arbeit von Nutzen sein kann, ist fraglich: In unstrukturiertem Gelände dominieren nicht-systematische Einflüsse; insbesondere die Unterscheidung zwischen Bodenkontakt und Rutschen ist kritisch im Hinblick auf die Odometrieberechnung von Robotern, die mehr als ein Bein gleichzeitig bewegen³. Deshalb verzichtet diese Arbeit auf die Modellierung systematischer Fehler.

An dieser Stelle bleibt noch anzumerken, dass die Odometrieberechnung in dieser Form für dynamisch laufende Roboter nicht anwendbar ist: Zum einen sind die Lageänderungen dieser Roboter während der Flugphasen nicht erfassbar, da zu deren Berechnung nur die Beine mit Kontakt zum Boden herangezogen werden. Zum anderen kann die Körperorientierung nur dann zuverlässig bestimmt werden, wenn mindestens drei Beine auf dem Boden stehen; befinden sich beispielsweise nur zwei Beine am Boden, kann der Zentralkörper theoretisch um die daraus gebildete Achse frei drehen (man rufe sich das Bild eines Reckturners ins Gedächtnis). Deshalb müssen derartige Roboter nicht gänzlich auf einen Koppelnavigationsansatz verzichten: in Verbindung mit den im nachfolgenden Abschnitt aufgeführten Inertialsystemen ist auch eine Lösung dieser Problematik denkbar.

3.2.1.2 Inertiale Navigationssysteme

Inertiale Navigationssysteme, oder kurz Inertialsysteme, werden schon seit langem im Flugwesen eingesetzt und gewinnen, auch in mobilen Robotersystemen, zunehmend an Bedeutung. Inertialsysteme bestehen aus Gyroskopen und Beschleunigungssensoren (Abbildung 3.2). Es werden Rotations- und Beschleunigungsraten gemessen und daraus Position, Geschwindigkeit und Orientierung geschätzt. Inertialsysteme lassen sich in zwei Kategorien einteilen:

Gimbaled-Systeme bestehen aus einer kardanisch aufgehängten, stabilisierten Plattform. Auf der Plattform installierte Beschleunigungsmesser sind unabhängig von

²Auf LAURON müssen keine zusätzlichen Sensoren angebracht werden. Alle für Berechnung der Odometrie notwendigen Sensoren werden von der reaktiven Laufmaschinensteuerung ebenfalls benötigt.

³AMBLER bewegt in der Regel nur eines seiner sechs Beine.

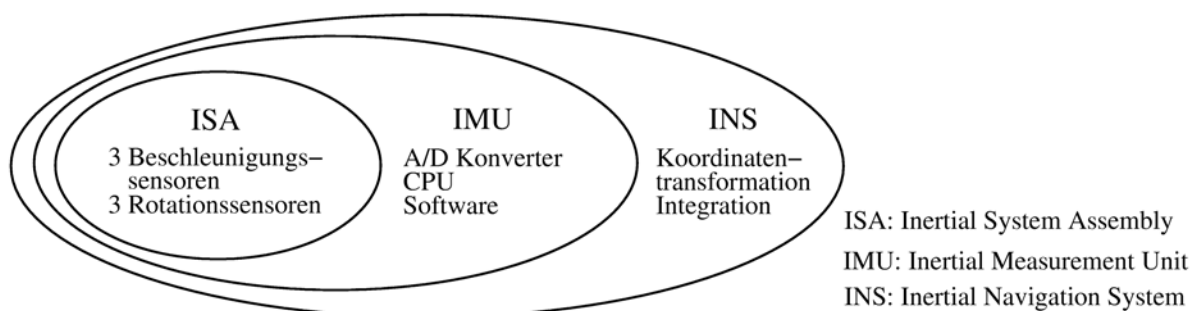


Abbildung 3.2: *Strapdown Inertial-Navigationssystem*: Die orthogonale Anordnung der drei Beschleunigungssensoren und der drei Gyroskope zu einem Sensorarray wird in der Literatur als *Inertial Sensor Assembly (ISA)* bezeichnet. Werden die Sensoren zusammen mit A/D-Wandlern und Signalverarbeitungsalgorithmen wie beispielsweise Filtern betrachtet, so wird diese Gesamtheit als *Inertial Measurement Unit (IMU)* bezeichnet. Unter einem *Inertialen Navigationssystem (INS)* versteht man schließlich eine IMU zusammen mit Algorithmen zur Fortschreibung von Orientierung und Position.

den Rotationen des Trägerfahrzeugs. Die zweifache Integration der Beschleunigungsmesswerte liefert somit direkt den in Nord-, Ost-, und Gravitations-Richtung zurückgelegten Weg.

Strapdown-Systeme werden hingegen direkt auf den Rahmen des Fahrzeugs montiert und erfahren alle Drehungen und Momente, die auf das Fahrzeug wirken.

Strapdown-Systeme sind wesentlich günstiger als die mechanisch sehr aufwändigen Gimbaled-Systeme. Abbildung 3.2 zeigt die Struktur eines Strapdown-Inertial-Navigationssystems (INS).

Inertialsysteme besitzen Nulloffsetfehler (die so genannten Biase), Skalierungsfehler und Temperaturabhängigkeiten. Zur Bestimmung der Position und Orientierung werden die Drehraten und Beschleunigungsmesswerte integriert. Der so genannte Strapdown-Algorithmus ist der hierfür verwendete Standardalgorithmus (Titterton und Weston, 1997). Durch die einfache bzw. zweifache Integration der Messwerte führen schon kleinste konstante Messfehler zu unbegrenzt wachsenden Fehlern. Die Lagebestimmung mittels Inertialsystemen ist deshalb nur über kurze Zeitintervalle genau.

Inertialsysteme benötigen im Gegensatz zur Odometrie keinerlei physischen Kontakt mit der Umgebung und können omnidirektionale Bewegungen in 6D erfassen. Dies macht sie für den Einsatz in unstrukturierter Umgebung sehr attraktiv. Barshan und Durrant-Whyte (1995) kommen jedoch zu dem Schluss, dass Positionsschätzungen mittels Beschleunigungssensoren auf Grund der doppelten Integration sehr fehleranfällig und auf mobilen Robotern fast unbrauchbar sind. Insbesondere bei geringen Beschleunigungen, wie beispielsweise bei langsamen Drehungen, ist der Rauschabstand (engl. signal-to-noise ratio) gering. Änderungen in der Bewegungsrichtung detektiert ein INS anhand der auftretenden Radialbeschleunigungen, wodurch diese kaum beobachtbar sind.

Häufig werden Inertialsysteme oder auch Gyroskope alleine zur Schätzung der Roboterorientierung herangezogen: Fuke und Krotkov (1996) reduzieren den Orientierungsfehler der Odometrie eines Mondfahrzeuges, indem sie mit Hilfe eines Kalman-Filters die Messungen des Gyroskops und der Beschleunigungssensoren fusionieren. Gyroskopmessungen sind gut für hochfrequente, also sich häufig ändernde, Bewegungen geeignet. Beschleunigungssensoren sind empfindlich für Störungen durch die in diesem Bereich auftretenden hochfrequenten Vibrationen, liefern jedoch bei niederfrequenten Bewegungen genaue Werte. Gyroskope und Beschleunigungssensoren ergänzen sich komplementär. Das Kalman-Filter fusioniert sie so, dass das Gyroskop bei hochfrequenten Bewegungen und der Beschleunigungsmesser bei niederfrequenten Bewegungen dominant ist.

3.2.1.3 Zusammenfassung

Zusammenfassend lässt sich feststellen, dass sich der Einsatz von Inertialsystemen auf langsam laufenden Robotern im Gelände als Odometrieersatz grundsätzlich problematisch gestaltet. Sowohl die Vibrationen beim Laufen als auch die niedrige Bewegungsgeschwindigkeit erfordern sehr hohe Ansprüche an die Beschleunigungssensoren. Es ist fraglich, ob teure, hochwertige Sensorsysteme diesen Ansprüchen genügen. Derzeit verfügbare Inertialsysteme können die Odometrie für Laufmaschinen im Gelände nicht ersetzen. Wie schon am Ende des letzten Abschnitts angedeutet, sind sie jedoch als Unterstützung der Odometrie denkbar: In Phasen schnellen Laufens können z.B. Flugphasen überbrückt oder beim Klettern ein Abrutschen des kompletten Roboters detektiert werden.

3.2.2 Absolute Orientierungsbestimmung

Die Roboterorientierung (α, β, γ) , auch als *Roll*, *Pitch* und *Yaw* bezeichnet, hat einen sehr großen Einfluss auf die Genauigkeit der Koppelnavigation (vergleiche Anhang A). In diesem Abschnitt werden einige Verfahren vorgestellt, die durch Messung natürlicher Größen Rückschlüsse auf die absolute Orientierung der mobilen Plattform ermöglichen.

3.2.2.1 Gravitation

Die Gravitation zeigt immer in Richtung des Erdmittelpunktes und somit nach unten in Richtung der negativen z -Achse. Bei der Verwendung von Beschleunigungssensoren in einem Inertialsystem muss beispielsweise die gemessene Gravitationsbeschleunigung herausgerechnet werden, um korrekte Positionsdaten zu erhalten. Die Erdgravitation kann zur Bestimmung der absoluten Roboterorientierung herangezogen werden: Die Komponenten Roll und Pitch (α, β) des Orientierungsvektors können bestimmt werden; eine Aussage über die Yaw-Komponente (γ) ist hingegen nicht möglich. Die Messung der Erdbeschleunigung (in unseren Breitengraden $9,81\text{m/s}^2$) stellt an die Beschleunigungssensoren keine größeren Ansprüche, so dass, im Gegensatz zum Einsatz in Inertialsystemen, auch einfache Sensoren dieser Aufgabe gerecht werden können. Da Beschleunigungssensoren jedoch empfindlich gegenüber hochfrequenten Bewegungen sind,

empfiehlt sich, wie im letzten Abschnitt beschrieben, die zusätzliche Nutzung von Gyroskopen zur Stabilisierung der Werte.

3.2.2.2 Sonnenstandsensoren

Sonnenstandsensoren verfolgen die Yaw-Komponente der Orientierung anhand des Sonnenstandes. Sie können nur unter freiem Himmel verwendet werden und kommen z.B. in der Seefahrt oder in planetaren Rovern zum Einsatz (vergleiche hierzu Abschnitt 2.3.2). Preiswerte Sonnenstandsensoren liefern Winkel mit einer Auflösung von 6° – 10° . Sie können jedoch nur verwendet werden, wenn die Sonne nicht durch Wolken, Gebäude, Bäume u.ä. verdeckt wird. Somit beschränkt ein Sonnenstandsensor die Aktionszeiträume eines Fahrzeugs auf der Erde zu stark, und wird daher meist gar nicht in Erwägung gezogen.

3.2.2.3 Kompass

Der Kompass misst die Richtung des Erdmagnetfeldes und gibt Auskunft über die Richtung des magnetischen Felds (z.B. die Nordrichtung) und somit der Yaw-Komponente des Orientierungsvektors. Die Bestimmung der dreidimensionalen Orientierung ist nicht möglich. Die Stärke und die Richtung des gemessenen Magnetfeldes werden durch elektrische und magnetische Felder verzerrt. Des Weiteren werden die Messungen durch Ansammlungen von ferromagnetischem Metall gestört. Lokale Verzerrungen des Magnetfeldes sind auch durch den Roboter selbst zu erwarten: Neben den im Roboter verbauten Metallen erzeugen die mitgeführten Rechereinheiten elektromagnetische Felder. Weiterhin ist nicht zu vergessen, dass Elektromotoren einen magnetischen Kern und eine Spule besitzen. Vor dem Einsatz eines Kompasses auf dem Roboter ist deshalb zu klären, ob durch geeignete Positionierung des Sensors in Verbindung mit einer Kalibrierung die gelieferten Daten nutzbar sind (vergleiche Abschnitt 7.1.3).

3.2.2.4 Zusammenfassung

Zusammenfassend lässt sich feststellen, dass Kompass und Gravitationsmessung komplementär sind und sich zur absoluten Orientierungsbestimmung in allen drei Komponenten ergänzen. Beide Verfahren sind im freien natürlichen Gelände einsetzbar und auf Laufmaschinen grundsätzlich anwendbar. Deshalb wird in dieser Arbeit der Einsatz eines entsprechenden Sensors zur Korrektur der odometrischen Orientierungsschätzung evaluiert (vergleiche Abschnitt 7.1.3).

3.2.3 Aktive Leuchtfeuer

Aktive Leuchtfeuer (engl. Active Beacons) ist der Oberbegriff für eine Reihe absoluter Ortsbestimmungsverfahren, die mit Hilfe von Trilateration oder Triangulation die Position im Raum bestimmen. Beide Verfahren arbeiten mit einem mobilen Empfänger (bzw. einem mobilen aktiven Sender) und mehreren aktiven Sendern (bzw. Empfängern), deren Position im Raum bekannt ist. Bei der Trilateration werden die Distanzen,

bei der Triangulation die Winkel zwischen Empfänger(n) und Sender(n) paarweise gemessen. Durch Betrachtung der geometrischen Zusammenhänge kann auf die Position, bei den Triangulationsverfahren auch auf die Orientierung, des mobilen Systems rückgeschlossen werden. Zur Bestimmung einer 2D-Position werden mindestens zwei, für die eindeutige Bestimmung einer 3D-Position mindestens drei Leuchttürme benötigt. Der mittlerweile wohl bekannteste Vertreter dieser Klasse der absoluten Verfahren, das nach der Trilateration arbeitende GPS, ist – ganz im Gegensatz zu den meisten anderen Verfahren dieser Art – im freien Gelände (zumindest auf der Erde) verfügbar und wird im Folgenden kurz eingeführt.

3.2.3.1 GPS

GPS steht für Global Positioning System und wurde im Rahmen des Navstar-Programms⁴ des US-Verteidigungsministeriums entwickelt. Es ermöglicht die präzise Positionsbestimmung eines Fahrzeugs oder einer Person. Das Navstar-GPS-System besteht aus einem Raumsegment, einem Kontrollsegment und einem Nutzersegment. Abbildung 3.3 illustriert den Systemaufbau. Das Raumsegment besteht aus über 24 Satelliten, die auf 6 verschiedenen Umlaufbahnen die Erde umkreisen. Die Umlaufbahnen wurden so gewählt, dass theoretisch immer vier Satelliten gleichzeitig sichtbar sind und somit eine dreidimensionale Lokalisation möglich ist. Das Kontrollsegment wird aus Kontroll- und Monitorstationen gebildet. Sie überwachen die Satellitenbahnen, berechnen gegebenenfalls Korrekturen und übermitteln diese an die Satelliten. Das Benutzersegment wird durch auf der Erde befindliche Empfänger, oft *Rover* genannt, gebildet.

Um eine dreidimensionale Positionsbestimmung vorzunehmen, werden die Signallaufzeiten vom Empfänger zu mindestens vier⁵ verschiedenen Satelliten gemessen. Auf Grund von Störungen sind die Messungen um einen bestimmten Betrag verfälscht. Man spricht deshalb von Pseudoentfernungen. Die GPS-Messungen werden beeinflusst durch satellitenbezogene Fehler, Signalausbreitungsfehler und empfängerbezogene Fehler. Satellitenbezogene Fehler entstehen durch Fehler in den Satellitenuhren und Satellitenbahnen. Signalausbreitungsfehler werden durch die Troposphäre und die Ionosphäre verursacht. Zur Kategorie der empfängerbezogenen Fehler gehören Empfängeruhrenfehler und Mehrwegefehler, die durch Mehrfachreflexionen des Signals auf dem Weg zum Empfänger verursacht werden.

Eine Verbesserung der Positionsschätzung kann mit dem so genannten *differential GPS (DGPS)* erreicht werden. Hierbei befindet sich ein Referenzempfänger an einer bekannten Position nahe der eigenen Position. Er erfährt dieselben Fehler wie der eigene Empfänger. Da seine Position genau bekannt ist, können Korrekturterme berechnet und übermittelt werden. Es lassen sich verschiedene Möglichkeiten des Zugangs zu DGPS-Daten unterscheiden:

Funk: Eine Referenzstation berechnet ihre lokalen Korrekturdaten und sendet diese

⁴Navstar steht für **N**avigational **S**atellite **T**iming and **R**anging.

⁵Theoretisch reichen drei Satelliten zur exakten Rekonstruktion der Position, allerdings besitzen normale GPS-Empfänger keine ausreichend präzise Uhr zum Abgleich mit den Satellitenuhren, so dass ein vierter Satellit benötigt wird.

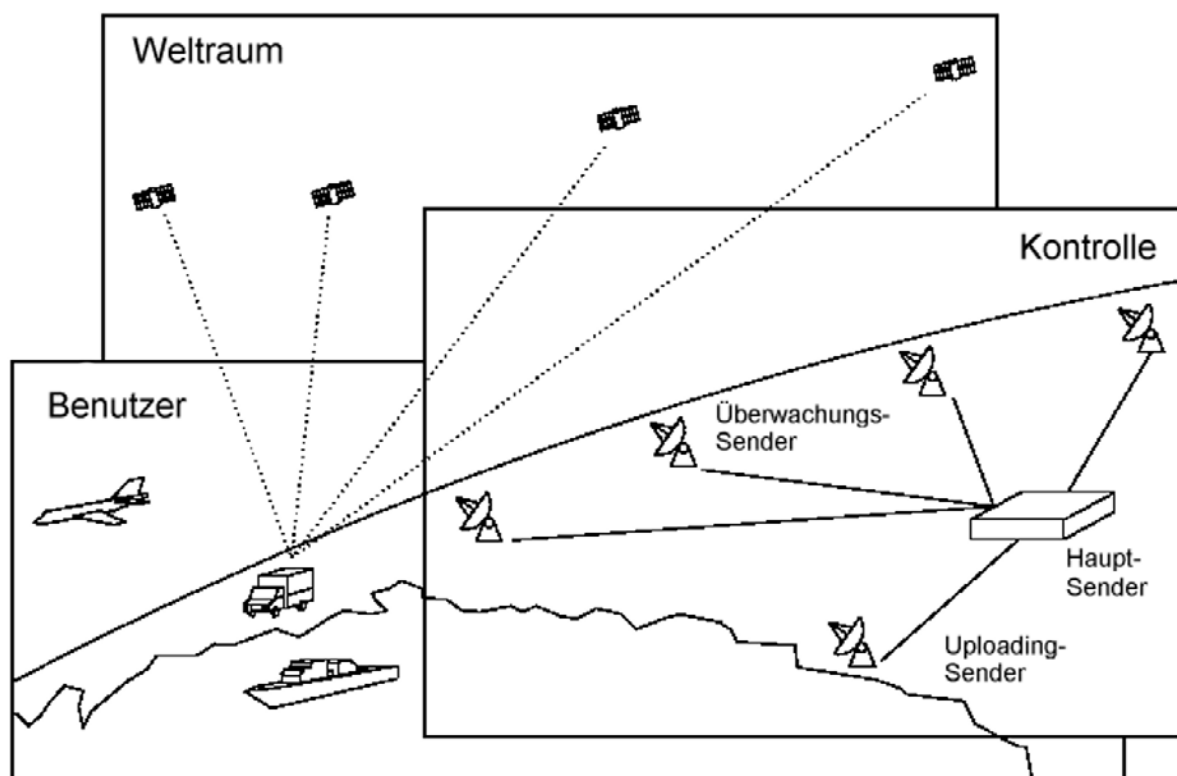


Abbildung 3.3: Das Navstar GPS-System besteht aus drei grundlegenden Segmenten: Raumsegment, Kontrollsegment und Benutzersegment (nach Borenstein et al. (1996)).

über Langwellensender, Mittelwellensender im Küstenbereich oder über das Radio Data Systems (RDS) im UKW-Frequenzbereich.

Satellitengestützt: Beim satellitengestützten DGPS (Satellite Based Augmentation System, SBAS) senden geostationäre Satelliten die Korrekturkarte eines größeren Gebietes, welche mit Hilfe eines Netzwerks von Monitorstationen (Abbildung 3.3) berechnet wird. Im Vergleich zu terrestrischen DGPS-Systemen ist die erreichbare Genauigkeit höher. Zu beachten ist, dass der Empfänger Sichtkontakt zu einem der relativ flach über dem Horizont stehenden geostationären Satelliten haben muss.

Goel et al. (1999) kombinieren Odometrie und Gyroskopmessungen mit GPS-Messungen, um die Vorteile sowohl der absoluten als auch der relativen Lokalisationsverfahren zu nutzen. Ein Kalman-Filter wird für die Filterung der Radgeschwindigkeiten und der Orientierung benutzt. Es zeigt sich, dass gut kalibrierte Odometrie in Verbindung mit Gyroskopdaten als Backup-System für kurzzeitige Verfolgung nutzbar ist.

Nebot et al. (1997); Sukkarieh et al. (1999) verwenden zur Koppelnavigation eines autonomen Landfahrzeuges ein Inertialsystem, dessen Positions-, Geschwindigkeits- und Orientierungsschätzung durch GPS-Daten korrigiert wird. Das dabei eingesetzte Kalman-Filter basiert auf einem Fehlermodell, das die Bias-Fehler des Inertialsystems berücksichtigt. Tritt ein Sprung in den GPS-Messungen auf, so führt dies zur fälschlichen Korrektur des Inertialsystems. Um dieses Problem zu lösen, wird überprüft, ob die

Typische Positionsgenauigkeit von GPS	± 15 Meter
Typische DifferentialGPS (DGPS)-Genauigkeit	$\pm 3\text{--}5$ Meter
Typische Genauigkeit mit aktiviertem SBAS	$\pm 1\text{--}3$ Meter
Typische RTK-Genauigkeit	$\pm 0,1$ Meter

Tabelle 3.1: Übersicht zu erwartender GPS-Genauigkeiten, Daten aus (Kowoma, 2004).

GPS-Daten zur angenommenen Grundverteilung passen. Ist dies nicht der Fall, werden die Daten abgelehnt und das Inertialsystem wird nicht korrigiert.

Insgesamt folgern Sukkariet et al. (1999), dass die Fusion von Inertialsystem- und GPS-Messungen gut für Navigationssysteme geeignet ist. Sie setzen dabei voraus, dass die Kovarianzmatrizen des Kalman-Filters entsprechend abgestimmt sind. Insbesondere sollen die Zustands- und Beobachtungskovarianzmatrizen so gewählt werden, dass das Inertialsystem nicht immer dem GPS-System folgt, vor allem dann nicht, wenn Sprünge auftreten. Die Kovarianzmatrizen dienen daher auch der Fehlererkennung. Modellieren die Kovarianzmatrizen eine zu große Unsicherheit in den Orientierungswerten, so kommt es zu oszillierenden Korrekturen.

Entsprechend der Analyse in Abschnitt 3.2.1.2 kommen Inertialsysteme für Laufmaschinen zwar nicht in Frage, jedoch steht dem Einsatz von GPS zur absoluten Positionskorrektur nichts entgegen. Nach Sukkariet et al. (1999) scheinen auch geeignete Methoden verfügbar, um der in Abschnitt 3.1 formulierten Anforderung an eine Positionsangabe möglichst ohne Sprünge gerecht zu werden. Tabelle 3.1 listet die erreichbaren Genauigkeiten verschiedener GPS-Systeme auf. Obwohl RTK⁶-GPS-Empfänger momentan noch sehr teuer sind, versprechen die aufgeführten Genauigkeiten mit der Inbetriebnahme des EGNOS⁷ für Mitteleuropa vielversprechende Daten.

3.2.4 Landmarken- und kartenbasierte Verfahren / SLAM

Landmarken sind unterscheidbare Umweltmerkmale, die ein Roboter mit Hilfe seiner Sensoren erkennen kann. Im Gegensatz zu den aktiven Leuchtfeuern sind Landmarken weder aktive Sender noch Empfänger. Merkmale, die ausschließlich zum Zweck der Roboterlokalisierung in der Umgebung platziert werden, wie beispielsweise Reflektoren oder Leitbänder, werden als *künstliche* Landmarken bezeichnet. Ihre Position ist im Allgemeinen bekannt und in einem Umweltmodell verfügbar. Zur leichteren Erkennbarkeit werden sie speziell auf die Umgebung abgestimmt; sie können zusätzliche Information enthalten, z.B. in Form von Barcodes (Tsumura, 1986). Werden in der Umgebung vorhandene Merkmale, wie beispielsweise geometrische Formen oder auch Deckenlampen

⁶Mit DGPS wird allgemein die Nutzung von Korrekturen der Pseudoentfernung bezeichnet. Stehen auch Korrekturdaten für die Trägerphasenmessungen zur Verfügung, die in Echtzeit gesendet werden, so spricht man von einem Real-Time Kinematic (RTK) System. Die Position kann im Idealfall auf wenige Zentimeter genau bestimmt werden. Die RTK-Korrekturdaten werden über GSM-Mobilfunk übertragen.

⁷Das amerikanische Wide Area Augmentation System (WAAS), das European Geostationary Navigation Overlay System (EGNOS) und der kommerzielle Omnistar-Dienst zählen zu den SBAS-Systemen.

(Launay et al., 2002), für die Lokalisation ausgenutzt, so handelt es sich um *natürliche* Landmarken. Die grundlegenden Schritte der landmarkenbasierten Lokalisation sind:

1. Aquirieren von Sensorinformationen
2. Detektieren und Segmentieren der Landmarken
3. Korrespondenzsuche im Umweltmodell
4. Berechnen der Position

Wird vor der Korrespondenzsuche eine lokale Karte aufgebaut, die anschließend mit einer globalen Karte verglichen wird, spricht man von einem kartenbasierten Ansatz. Der Übergang von natürlichen Landmarken zu kartenbasierten Verfahren verläuft fließend: Viele landmarkenbasierte Verfahren legen das Umweltmodell in einer Art Karte ab. Im Gegenzug nehmen viele kartenbasierte Verfahren natürliche Landmarken, wie beispielsweise Ecken oder Kanten, zur Grundlage (Cox, 1991). Andere verzichten wiederum auf die explizite Extraktion von Landmarken aus den Sensordaten (Zhang und Faugeras, 1992).

Oftmals kommen probabilistische Ansätze zum Einsatz, welche die Wahrscheinlichkeit $p(l|s, a)$ bestimmen, den Roboter unter den gegebenen Sensormessungen s und den (meist odometrischen) Bewegungsdaten a in der Lage l zu finden. Die „klassische“ probabilistische Methode, um Unbestimmtheiten zur Lageverfolgung zu nutzen, ist das erweiterte Kalman-Filter (Leonard und Durant-White, 1991a). Als Wahrscheinlichkeitsverteilung $p(l|s, a)$ wird eine einfache Gauß-Verteilung angenommen. Dies hat den Vorteil, dass die Roboterlage präzise bestimmt werden kann. Diese Verfahren setzen voraus, dass vor der Korrespondenzsuche die ungefähre Lage des Roboters verfügbar ist, was wiederum die Berechnung effizient macht. Die größten Nachteile dieser Verfahren liegen darin, dass sie sich von größeren Lokalisationsfehlern in der Regel nicht mehr erholen können und auch Mehrdeutigkeiten Probleme bereiten. Um diesen Nachteil auszuräumen, geht man dazu über, mit mehreren erweiterten Kalman-Filtern gleichzeitig mehrere Positionshypothesen zu verfolgen (Jensfelt und Kristensen, 2001).

Die Markov-Lokalisation (Fox, 1998) wählt hingegen eine explizite diskrete Repräsentation der Wahrscheinlichkeitsverteilung $p(l|s, a)$ über alle möglichen Roboterlagen in einem Gitter oder topologischen Graphen. Hierdurch wird es möglich, den Roboter zu lokalisieren, auch wenn die initiale Lage unbekannt ist. Mehrdeutigkeiten können repräsentiert und aktiv aufgelöst werden. Markov-Lokalisationsverfahren können mit verrauschten Sensordaten, wie beispielsweise Ultraschall, umgehen. Der große Nachteil der Markov-Lokalisation ist deren hoher Berechnungsaufwand. Aus diesem Grund etablierten sich die Monte-Carlo-Verfahren, auch Partikelfilter genannt (Dellaert et al., 1999). Diese approximieren die Wahrscheinlichkeitsverteilung $p(l|s, a)$ durch eine Menge gewichteter Tupel $\langle l, w \rangle$, wobei l eine Lage und w ein Maß der Bedeutung der Lage ist. Je bedeutungsvoller eine Lage desto wahrscheinlicher wird diese in der nächsten Stichprobe enthalten sein. Somit wird eine explizite Berechnung der gesamten Wahrscheinlichkeitsverteilung vermieden.

Ist eine a-priori-Karte nicht verfügbar, so muss diese simultan zur Lokalisation aufgebaut werden. Zurückgehend auf Leonard und Durant-White (1991b) in der Fachliteratur kurz als SLAM (engl. Simultaneous Localisation And Mapping) bezeichnet, stellt dies gewissermaßen das „Henne-Ei-Problem“ der Lokalisation dar und gilt als eine der großen Herausforderungen der mobilen Robotik: Ohne eine Karte kann sich der Roboter nicht lokalisieren und ohne bekannte Roboterposition kann keine Karte aufgebaut werden. Die meisten SLAM-Algorithmen lassen sich einer der folgenden Klassen zuordnen (Russell und Norvig, 2004):

Kalman-Filter-Ansätze stellen die Karte der Umgebung als Menge von Landmarken dar. Für jede Landmarke verwalten sie die geschätzte Position und die Unsicherheit, mit welcher die Schätzung verbunden ist. Mit jeder Sensormessung schätzen sie die Positionen aller Landmarken und des Roboters so, dass diese den kleinsten quadratischen Abstand von den realen Positionen besitzen.

Belegtheitsgitter-Verfahren kombinieren die Erstellung eines Belegtheitsgitters mit einem Lokalisationsverfahren (vergleiche hierzu Abschnitt 4.3). Die Neuberechnung der Karte und die Positionskorrektur der Lokalisation werden im gegenseitigen Wechsel ausgeführt.

Expectation-Maximization-Verfahren führen in einem so genannten Erwartungsschritt eine Form der Markov-Lokalisation durch, um im anschließenden Maximierungsschritt die wahrscheinlichste Karte zu berechnen.

Relaxations-Algorithmen berechnen die optimale Karte nur mit Hilfe von relativen Odometriemessungen. Die Karte wird als attributierter Graph dargestellt: Landmarken mit Position und Unsicherheit entsprechen den Knoten. Die Kanten, mit Distanz- und Winkelangaben versehen, repräsentieren die gefahrenen Routen zwischen den Landmarken. Ein zusätzliches Konfidenzmaß der Kanten gibt an, mit welcher Wahrscheinlichkeit der Roboter diese Route befahren kann.

Bei der simultanen Lokalisation und Kartierung beeinflussen sich die Verfahren untereinander, so dass die Fehler der verschiedenen Sensormessungen in der Regel nicht mehr als voneinander unabhängig betrachtet werden können. Je nach Repräsentation der Karte weist die SLAM-Problematik weiterhin eine sehr hohe Dimensionalität auf, was den Suchraum und somit auch die Komplexität der Kartenberechnung enorm erhöht. Darüber hinaus muss der Roboter das Korrespondenzproblem lösen: Gehören Sensormessungen, die zu verschiedenen Zeitpunkten aufgenommen wurden, zum gleichen Objekt der Umgebung, oder nicht? Schließlich führt die Dynamik der Umgebung dazu, dass Sensormessungen, die vom erwarteten Wert abweichen, mehrere Annahmen zulassen: Hat sich die Umgebung geändert, oder ist die Annahme für die Roboterposition fehlerbehaftet?

Die Positionsgenauigkeit von landmarken- und kartenbasierten Verfahren hängt im Allgemeinen von der Distanz zwischen dem Roboter und der Landmarke ab. Ist der Roboter weit weg von den Landmarken, ist die Positionsgenauigkeit gering. Beispielsweise lokalisieren Cozman et al. (2000) basierend auf dem Ansatz von Stein und Medioni

(1995) einen planetaren Rover anhand des *panoramischen Horizonts*, womit sie eine Genauigkeit im Bereich von hundert Metern erreichen. Höhere Genauigkeiten werden nur erreicht, wenn der Roboter sich nahe bei einer Landmarke befindet.

Obwohl der Begriff der „natürlichen“ Landmarken einen Bezug zur Natur nahe legt, ist festzustellen, dass von Menschenhand geschaffene Landmarken, wie sie vor allem in stark strukturierter Umgebung zu finden sind, viele Vorteile aufweisen: Einfache Geometrien sind leicht segmentierbar und lassen sich gut voneinander unterscheiden. Die nahezu konstanten Umgebungsbedingungen in Gebäuden ermöglichen darüber hinaus eine hohe Zuverlässigkeit bei der Sensordatenaquise. So stellen z.B. Karlsson et al. (2005) einen kommerziell verfügbaren, kamerabasierten, auf Grauwertmerkmalen basierenden SLAM-Ansatz vor, mit dem sich Roboter in Gebäuden zurechtfinden können.

Im Gegensatz hierzu stellen sich ändernde Lichtverhältnisse für kamerabasierte Verfahren oft weitreichende Probleme dar: Landmarken werden gar nicht mehr erkannt, oder andere Objekte mit ähnlichen Merkmalen werden irrtümlich für eine Landmarke gehalten. Trotzdem sind kamerabasierte Verfahren für die Lokalisation im Gelände geeignet: Bilder liefern großflächig Daten mit Helligkeits- und Farbinformationen, so dass die Extraktion von Landmarken z.B. auch auf einer weitläufigen Wiese grundsätzlich denkbar ist (Todt und Torras, 2000). Derartigen Umgebungen ist mit anderen Sensorensystemen, wie Laserscannern oder gar Ultraschall, nur schwer beizukommen.

In natürlichen, unstrukturierten Umgebungen ist es schwieriger, eindeutig identifizierbare und wiederauffindbare Merkmale als Landmarke zu klassifizieren. Des Weiteren stellt ein Verfahren, das in unterschiedlichen Umgebungen arbeiten kann, eine große Herausforderung für Sensorik und Algorithmen dar. Es sollte fähig sein, mit unterschiedlichen Typen von Landmarken umgehen zu können, um den hohen Ansprüchen gerecht zu werden. Dies deutet darauf hin, dass landmarken- und kartenbasierte Verfahren zur Lokalisation im Gelände ein äußerst komplexes Themengebiet darstellen.

Zur Anwendung auf einer Laufmaschine kommt erschwerend hinzu, dass die meisten landmarken- und kartenbasierte Verfahren – allen voran die große Gruppe der kamerabasierten Verfahren – rechenintensiv sind und somit die grundlegende Randbedingung der *Rechnereffizienz* aus Abschnitt 2.1 verletzen. Die Tatsache, dass im Gelände keine a-priori-Karte verfügbar ist, erfordert einen SLAM-Ansatz.

Landmarken- und kartenbasierte Verfahren als Ergänzung von GPS sind prinzipiell geeignet, um die Lokalisation im Gelände robust und zuverlässig zu machen: In freien offenen Gebieten sind kaum geeignete Landmarken zu finden; hier verspricht das GPS bei freier Sicht gute Positionswerte. In Bereichen mit verdeckter Sicht zu den Satelliten können die das GPS störenden Objekte als Landmarken zur Lokalisation dienen.

3.3 Schlussfolgerungen

Unter den gegebenen Randbedingungen ist keines der vorgestellten Verfahren alleine in der Lage, eine Laufmaschine im Gelände auf Dauer zu lokalisieren; eine Kombination erscheint durchaus sinnvoll. Tabelle 3.2 fasst die Bewertung der vorgestellten Lokalisationsverfahren hinsichtlich der in Abschnitt 3.1 gestellten Anforderungen zusammen.

	Landmarken- und kartenbasiert	GPS	Kompass und Gravitation	Inertialsystem	Odometrie
Allgemein:					
einfache Sensorik	--	++	++	++	++
Rechnereffizienz, Echtzeitfähigkeit	--	++	++	++	++
Lagefehler begrenzt	+	++	++	--	--
keine Positionssprünge	o	o	++	++	++
Unstrukturiertes Gelände:					
unabhängig von a-priori-Karten	o	++	++	++	++
unabhängig von strukturierten Umweltmerkmalen	o	+	+	++	++
robust gegen wechselnde Umgebungsbedingungen	-	+	+	++	++
robust gegen schwierige Bodenbedingungen	++	++	++	++	-
Laufmaschinen:					
sechs Freiheitsgrade bei der Bewegung	+	--	--	++	+
punktuellem Bodenkontakt	++	++	++	++	+
viele Bewegungsfreiheitsgrade	+	++	+	+	+
langsame Bewegung	++	+	++	--	++

++: gut geeignet +: geeignet o: mäßig geeignet -: wenig geeignet --: ungeeignet

Tabelle 3.2: Beurteilung der untersuchten Lokalisationsverfahren anhand der an sie gestellten Anforderungen: Kein Verfahren genügt, um eine Laufmaschine über längere Zeit hinweg alleine zu lokalisieren; eine Kombination aus relativen und absoluten Verfahren ist notwendig. Inertialsysteme scheitern insbesondere an den langsamen Bewegungen von Laufmaschinen. Trotz reduzierter Genauigkeit im Gelände stehen odometrische Daten zur Verfügung. Landmarken- und kartenbasierte Ansätze erfordern komplexe, rechenintensive Sensoren. In Kombination erfassen Kompass, Gravitation und GPS alle sechs Bewegungsfreiheitsgrade.

Der Lagefehler der beiden relativen Verfahren zur Koppelnavigation, Odometrie und Inertialsystem, kann stark anwachsen. Auf Grund der auftretenden langsamen Bewegungen kommt ein Inertialsystem aber nur zur Unterstützung der Odometrie in Frage. Die Odometrie kann für Laufmaschinen durchaus berechnet werden, obwohl sich die erzielbare Genauigkeit insbesondere unter schwierigen Bodenbedingungen in Grenzen hält. Der Nutzung als ständig verfügbares Basissystem zur Lokalisation, wie sie auf vielen radgetriebenen Plattformen praktiziert wird, steht aber nichts im Wege.

Die Begrenzung des Lagefehlers kann mit Hilfe absoluter Verfahren geschehen. Ein nichtblockierendes System zur absoluten Orientierungsbestimmung steht durch einen Kompass in Verbindung mit Beschleunigungssensoren zur Messung der Gravitation zur Verfügung. Kommerziell erhältliche Systeme nutzen u.a. Gyroskope, um die Orientierungswerte während der Bewegung zu stabilisieren (vergleiche Abschnitt 7.1.3); die Robustheit der Messungen gegenüber störenden Umwelteinflüssen bleibt noch zu untersuchen.

Des Weiteren sind im Gelände (auf der Erde) GPS-Daten in der Regel verfügbar, so dass dessen absolute Positionsinformationen verwendet werden können, um die absoluten Orientierungsinformationen zu einer absoluten 6D-Lagebeschreibung zu ergänzen. Insbesondere das für Mitteleuropa in Betrieb gehende EGNOS erzeugt gut verwertbare Informationen mit ausreichender Genauigkeit. Andererseits besteht nicht ständig Sichtkontakt zu den Satelliten wie beispielsweise im Wald, unter Brücken oder in Tunnels, so dass eine absolute Positionskorrektur nicht immer garantiert werden kann.

Für eine robustere Gestaltung der Lokalisation können landmarken- und kartenbasierte Ansätze herangezogen werden. Diese erfordern jedoch eine große Rechenleistung, so dass zusätzliche Rechnerhardware erforderlich wird. Der Einsatz derartiger Verfahren wird weiterhin durch fehlende a-priori-Karten, wechselnde Umgebungsbedingungen sowie unstrukturiertes Gelände erschwert. Daher wird im Folgenden auf eine detaillierte Untersuchung landmarken- und/oder kartenbasierter Lokalisationsansätze verzichtet.

Somit fällt die Wahl auf die am besten geeigneten Lokalisationsverfahren: Odometrie, Kompass & Gravitation sowie GPS. Zu beantworten bleibt noch die Frage: Wie können die Lagedaten dieser unterschiedlichen Verfahren miteinander fusioniert werden?

Sensordatenfusion bezeichnet hierbei den Prozess, durch den Daten unterschiedlicher Systeme in eine einheitliche Darstellungsform gebracht werden. Um die Lage eines Roboters anhand von Sensormessungen unterschiedlicher Genauigkeit zu bestimmen, werden Sensordatenfusionsschemata benutzt, die die Messungen möglichst optimal kombinieren. Sensordatenfusion kann auf verschiedenen Ebenen stattfinden. Die unterste Ebene ist durch die Signalebene, die oberste Ebene ist durch die Symbolebene gegeben. Fusionsmethoden lassen sich in numerische und geometrische Methoden unterteilen. Numerische Datenfusion beschreibt die direkte Verknüpfung von Messdaten. Bei der geometrischen Datenfusion werden die Messwerte in eine Darstellung integriert, die dann bei Bedarf interpretiert wird. Ein Beispiel hierfür wäre ein geometrisches Umweltmodell.

Die Antwort auf die gestellte Frage wurde in den letzten Abschnitten anhand etlicher Beispiele angeführt: Das Kalman-Filter ist eine gut verstandene und zur Lokalisation

oft eingesetzte numerische Fusionsmethode, die im System vorhandene Unbestimmtheiten ausnutzt, um statistisch optimale Ergebnisse zu liefern. Es wird zur Fusion der unterschiedlichen Sensordaten im Folgenden herangezogen.

Die weiteren Abschnitte dieses Kapitels beschreiben den in dieser Arbeit entwickelten Ansatz zur Lokalisation von Laufmaschinen im Gelände.

3.4 Odometrie von Laufmaschinen

Wie schon in Abschnitt 3.2.1.1 erläutert, können auf einer Laufmaschine die Standardverfahren zur Odometrieberechnung radgetriebener Roboter nicht eingesetzt werden. Trotzdem soll die Odometrie nach Roston und Krotkov (1992) als relativ arbeitendes Basissystem zur Lagebestimmung dienen, da sie jederzeit verfügbar ist und trotz schwieriger Bedingungen im Gelände dennoch den grundlegenden Trend der Bewegung in allen sechs Dimensionen nachvollziehen kann.

3.4.1 Odometrieberechnung

Die im Folgenden skizzierte Berechnung der Koppelnavigation von Laufmaschinen nutzt sowohl Gelenkencoder der Beine als auch Kontaktsensoren in den Füßen, um Rückschlüsse auf die Zentralkörperbewegung zu ziehen: Die Stemmbeine einer Laufmaschine stehen im Kontakt zum Untergrund; sie tragen das gesamte Körpergewicht und schieben den Roboter voran. Erreichen sie die so genannte PEP (Posterior Extreme Position, hintere Extremposition) werden sie angehoben und gehen in die Schwingphase über. Ohne Einfluss auf die Zentralkörperbewegung werden sie zur so genannten AEP (Anterior Extreme Position, vordere Extremposition) geschwungen, um dort eine neuerliche Stemmphase zu beginnen.

Dementsprechend führen ausschließlich die Bewegungen der Stemmbeine zu einer Bewegung des Roboters in Bezug auf seine Umwelt. Unter Idealbedingungen bleibt die Position \mathbf{b}_j^g eines Stemmbeins j im globalen Koordinatensystem während der Stemmphase konstant; im Roboterkoordinatensystem führt es jedoch eine Bewegung aus, so dass sich die Position \mathbf{b}_j^r ständig ändert. Über eine Koordinatentransformation mit dem Translationsvektor \mathbf{p}^g und der 3×3 Rotationsmatrix R_r^g sind beide miteinander verknüpft:

$$\mathbf{b}_j^g = R_r^g \mathbf{b}_j^r + \mathbf{p}^g \quad . \quad (3.1)$$

Dabei sind die Beinpositionen \mathbf{b}_j^r im Roboterkoordinatensystem durch die Roboterkinematik und die Stellung der Gelenkwinkel stets bekannt; die Positionen \mathbf{b}_j^g im globalen Koordinatensystem sind durch den zeitlichen Verlauf bestimmt. Durch die Rotationsmatrix R_r^g und den Translationsvektor \mathbf{p}^g ist die Lage des Koordinatenursprungs des Roboterkoordinatensystems r bzgl. des globalen Koordinatensystems g und damit die gesuchte Roboterlage L eindeutig bestimmt (vergleiche hierzu Anhang A.2):

$$L = (\mathbf{p}^g, \boldsymbol{\psi}^g) \quad , \quad (3.2)$$

wobei die Orientierung $\boldsymbol{\psi}^g$ sich nach Anhang A.2 direkt aus der Rotationsmatrix R_r^g berechnen lässt.

Nun gilt es, das durch alle Stemmbeine aufgestellte Gleichungssystem nach R_r^g und \mathbf{p}^g aufzulösen. Diese sind eindeutig bestimmt, wenn drei nicht kollineare Stemmbeine betrachtet werden. Sind weniger als drei Stemmbeine verfügbar, kann der Roboter theoretisch frei um die durch die zwei Fußaufsetzpunkte gebildete Gerade (bzw. den einen Fußaufsetzpunkt) rotieren. Bei mehr als drei Stemmbeinen ist das Problem überbestimmt und auf Grund verrauschter Sensordaten sowie Modellunsicherheiten keine eindeutige Lösung mehr zu erwarten, weshalb eine Lösung mit Hilfe der kleinsten Fehlerquadrate berechnet werden kann. Der quadratische Fehler F über alle Stemmbeine j soll minimiert werden:

$$F(R_r^g, \mathbf{p}^g) = \sum_j g_j \mathbf{f}_j^T \mathbf{f}_j \quad \rightarrow \quad \min \quad , \quad (3.3)$$

mit Fehlern

$$\mathbf{f}_j = \mathbf{b}_j^g - R_r^g \mathbf{b}_j^r - \mathbf{p}^g \quad (3.4)$$

und auf Beobachtungen basierenden Gewichten

$$g_j = \begin{cases} 0 & \text{Bein } j \text{ befindet sich in der Schwingphase} \\ 1 & \text{Bein } j \text{ befindet sich in der Stemmphase} \end{cases} \quad (3.5)$$

Zur Lösung der Gleichung 3.3 kann auf die so genannte *singuläre Wertzerlegung* (engl. singular value decomposition) zurückgegriffen werden, welche im Folgenden kurz erläutert wird. Es seien:

$$g = \sum_j g_j \quad , \quad (3.6)$$

$$\mathbf{b}^g = \sum_j g_j \mathbf{b}_j^g \quad , \quad (3.7)$$

$$\mathbf{b}^r = \sum_j g_j \mathbf{b}_j^r \quad , \quad (3.8)$$

$$B = \sum_j g_j \mathbf{b}_j^g \mathbf{b}_j^{rT} \quad (3.9)$$

und

$$E = B - \frac{1}{g} \mathbf{b}^g \mathbf{b}^{rT} \quad , \quad (3.10)$$

wobei

$$USV^T = E \quad (3.11)$$

die *singulären Wertzerlegung* von E ist. Dann sind

$$\begin{aligned} \hat{R}_r^g &= UV^T \quad , \\ \hat{\mathbf{p}}^g &= \frac{1}{g} (\mathbf{b}^g - \hat{R}_r^g \mathbf{b}^r) \end{aligned} \quad (3.12)$$

die besten Schätzungen für R_r^g und p^g , welche die Gleichung 3.3 minimieren.

Nachdem ein Berechnungsschritt durchgeführt wurde, muss noch die globale Position der Beine aktualisiert werden:

$$\mathbf{b}_j^g = \begin{cases} \hat{R}_r^g \mathbf{b}_j^r + \hat{p}^g & \text{Bein } j \text{ befindet sich in der Schwingphase} \\ \mathbf{b}_j^g & \text{Bein } j \text{ befindet sich in der Stemmphase} \end{cases} \quad (3.13)$$

3.4.2 Gewichtung der Beine

Die erzielbare Genauigkeit dieser Methode im Gelände hängt neben einer gut kalibrierten Roboterkinematik in erster Linie von einer geschickten Wahl der Gewichte g_j ab. Die einfache Formulierung aus Gleichung 3.5 wird störenden Einflüssen in realen Umgebungen nicht gerecht:

1. Ein auf dem Boden stehendes Bein kann z.B. auf weichem Untergrund einsinken oder auf Grund innerer Verspannungen wegrutschen: Somit bewegt sich das Bein im globalen Koordinatensystem; es verletzt die Annahme einer konstanten Position \mathbf{b}_j^g während der Stemmphase.
2. Der Übergang von Schwing- zu Stemmphase ist in der Regel fließend: Der Fuß berührt den Boden, wird dann allmählich belastet, bis er schließlich fest auf dem Boden steht. Nicht jede Beinbewegung während dieser Phase trägt zur Bewegung des Zentralkörpers bei. Wann genau ist der Zeitpunkt für den Phasenwechsel anzusetzen?
3. Durch unebenen Untergrund kann der gesamte Roboter kippen oder wegrutschen, ohne dass sich überhaupt ein Bein bewegt.

Dem ersten Punkt könnte durch eine einfache Überprüfung auf rutschende Beine, die anschließend bei der Berechnung ignoriert werden, entgegnet werden. Jedoch scheitert dieses Vorgehen bei Laufmaschinen wie beispielsweise LAURON, die gleichzeitig mehrere Beine schwingen können: Beim Tripod, der „Standard“-Gangart sechsbeiniger Insekten wie z.B. bei Stabheuschrecken, sind in der Regel nur drei Beine gleichzeitig auf dem Boden. Rutscht eines, wären nur noch zwei Beine für die Odometrieberechnung verwendbar, wodurch die Roboterlage nicht mehr eindeutig bestimmbar ist. Somit bedarf es einer differenzierten Betrachtung rutschender Beine. Der zweite Punkt erfordert darüber hinaus einen „weichen“ Übergang von Schwing- zu Stemmphase. Durch die Odometrie nicht erfassbar ist der dritte und letzte Punkt: Dieser Art von Fehler kann nur durch Hinzunahme weiterer Sensorik erfasst werden.

Den beiden ersten Punkten kann durch die Wahl kontinuierlicher Werte für die Gewichte $g_j \in [0; 1]$ Rechnung getragen werden. Doch welche Faktoren beeinflussen das Gewicht g_j , mit dem ein Bein j in die Odometrieberechnung einfließt?

- Schwingbeine haben überhaupt keinen Einfluss.
- Fest auf dem Boden stehende Beine bringen ihre Bewegung voll ein.

- Je stärker ein Bein auf dem Boden steht, desto weniger wahrscheinlich ist ein seitliches Rutschen in genau diesem Moment, denn während des Rutschens trägt das Bein in der Regel weniger Gewicht.
- Beine mit leichtem Bodenkontakt setzen unter Umständen gerade auf dem Untergrund auf, sinken ein oder rutschen zur Seite.

Diese Aufzählung lässt erkennen, dass der Grad des Bodenkontakts eines Beines eine gute erste Schätzung der Gewichte g_j darstellt. Da ein exaktes mathematisches Modell nicht formulierbar ist, bieten sich Methoden der Unscharfen Logik (*Fuzzy-Logik*) an, die in Anhang B kurz eingeführt werden. Diese ermöglichen es, das linguistische Erfahrungswissen in die Berechnung einzubringen: Das Wissen wird in Form von Regeln abgelegt und für die *Fuzzy-Implikation* verwendet. Ein weiteres Problem besteht darin, Aussagen wie beispielsweise „fest auf dem Boden stehend“ aus den zur Verfügung stehenden Sensorwerten zu quantifizieren. Die Modellierung durch die linguistische Terme „wenig“, „mittel“, „viel“ der linguistischen Variablen „Fußkraft“ ist für den Anwender gut nachvollziehbar. Jedoch müssen auch hierbei die Wertebereiche der linguistischen Terme bestimmt werden.

Folgende Gründe sprechen für die Nutzung von Fuzzy-Logik:

- Nichtlineare Zusammenhänge lassen sich leicht formulieren.
- Die Regeln lassen sich leicht und verständlich erstellen. Implikationen wie „Wenn A, dann B“ lassen sich, im Gegensatz zu einer parametrisierten mathematischen Gleichung, direkt formulieren und verarbeiten, wie sie wahrgenommen werden. Auch sind die Zusammenhänge direkt aus den Regeln ablesbar.
- Der Regelsatz lässt sich auf einfache Weise erweitern, indem man weitere Regeln hinzufügt. Dabei verlassen die Ausgangsgrößen ihren Definitionsbereich nicht.
- Die Regeln können auf andere Roboterarchitekturen oder andere Sensorik übertragen und angepasst werden. Dazu genügt es, wenn die entsprechenden Eingangs-Fuzzy-Mengen angepasst werden.
- Fuzzy-Methoden tragen zur Reduktion unerwünschter Auswirkungen nicht perfekter Modelle bei. Nicht immer wird man in der Lage sein, vollkommen konsistente Regelsätze zu erstellen. Fuzzy-Methoden modellieren per Definition Unschärfen: Widersprüchliche Regeln werden toleriert, ihre Effekte heben sich auf.

Wie bereits angedeutet, bilden vor allem die Kräfte, die auf die einzelnen Füße des Roboters wirken, die Grundlage für eine unscharfe Bewertung der Beingewichte g_j . Neben dem Absolutbetrag der einzelnen Fußkräfte F_j^a mag beispielsweise auch der Anteil der Fußkraft an der Gesamtkraft, also der relative Kraftanteil F_j^r eine vielversprechende Wahl als linguistische Variable in der Modellierung des Fuzzy-Bewerters darstellen, um der unterschiedlichen Beinanzahl am Boden im Grenzbereich gerecht zu werden. Dieser

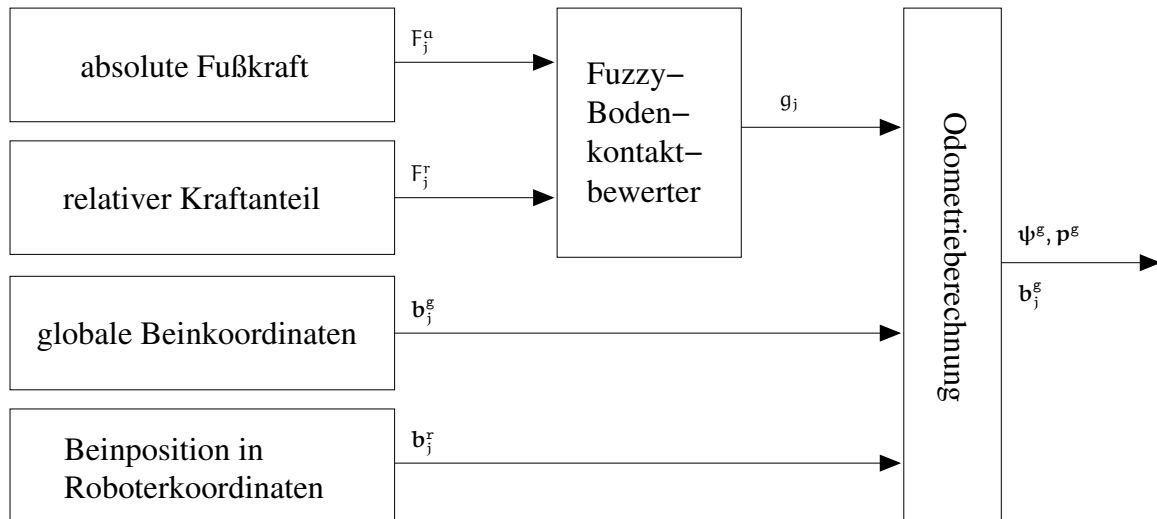


Abbildung 3.4: Ablauf der Odometrieberechnung von Laufmaschinen: Der Fuzzy-Bodenkontaktbewerter berechnet die Beingewichte g_j aus den absoluten (F_j^a) und relativen (F_j^r) Fußkräften, die zusammen mit den roboter-lokalen (\mathbf{b}_j^r) und den globalen (\mathbf{b}_j^g) Beinkoordinaten in die Odometrieberechnung eingehen.

nutzt schließlich Regeln der folgenden Form:

$$\begin{array}{ll}
 \text{WENN } F_j^a = \text{sehr hoch} & \text{DANN } g_j = \text{sehr hoch} \\
 \text{WENN } F_j^a = \text{sehr niedrig} & \text{DANN } g_j = \text{sehr niedrig} \\
 \text{WENN } F_j^a = \text{niedrig UND } F_j^r = \text{hoch} & \text{DANN } g_j = \text{mittel} \\
 \text{WENN } F_j^a = \text{mittel UND } F_j^r = \text{niedrig} & \text{DANN } g_j = \text{niedrig}
 \end{array} \quad (3.14)$$

Der komplette Regelsatz, der im Rahmen der vorliegenden Arbeit entwickelt wurde, findet sich in Anhang C.1. Abbildung 3.4 skizziert die vorgeschlagene Odometrie-berechnung.

Durch die Einführung kontinuierlicher Gewichte g_j verletzen rutschende Beine die Grundannahme fixierter Stemmbeine. Wie soll die Aktualisierung der Beinpositionen nach einem Odometrie-Berechnungsschritt durchgeführt werden? Wird ein Bein fälschlicherweise als fixes Stemmbein behandelt, verschiebt sich der Zentralkörper entgegen dessen Rutschrichtung. Wird ein Bein fälschlicherweise als Schwingbein interpretiert, kann bei weniger als 3 verbleibenden Stemmbeinen die Odometrie-berechnung nicht durchgeführt werden. Die Beinbewegung in Rutschrichtung geht verloren. Ein Kompromiss zwischen den beiden Ansätzen verfolgt das folgende Vorgehen: Das Bein wird teilweise als Schwing- und teilweise als Stemmbein betrachtet. Die alten und neuen Beinkoordinaten werden mit Hilfe des Faktors g_j gewichtet. Die Gleichung 3.13 zur Aktualisierung der Beinposition wird ersetzt durch die folgende:

$$\mathbf{b}_j^g = (1 - g_j)(\hat{R}_r^g \mathbf{b}_j^r + \hat{\mathbf{p}}^g) + g_j \mathbf{b}_j^g \quad (3.15)$$

Ein weiterer Lösungsansatz wäre, die globalen Koordinaten *aller* Beine nach der Lageberechnung neu zu berechnen. So einfach und rechnerisch korrekt diese Lösung auch scheint, in der Praxis führen kleinste Motorbewegungen schon im Stand zu einem vermeintlichen Driften der Lage.

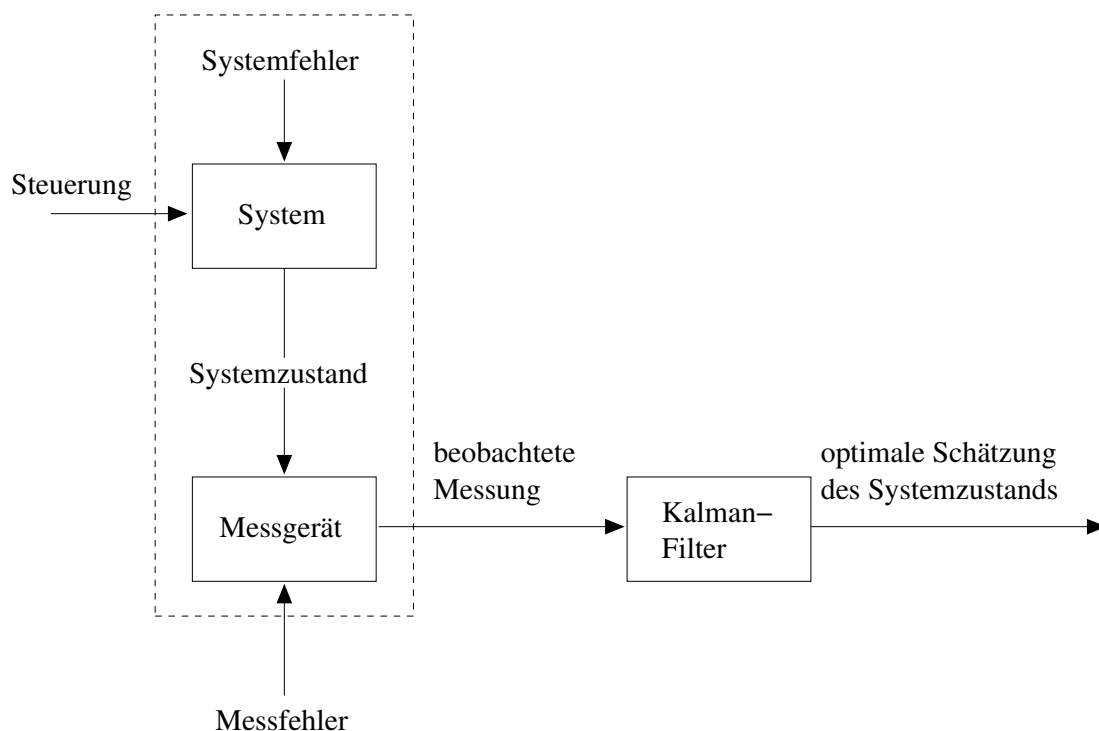


Abbildung 3.5: Typische Anwendung des Kalman-Filters, nach Maybeck (1979).

3.5 Das Kalman-Filter

Um den bei der Odometrie entstehenden Lagefehler zu begrenzen, werden in dieser Arbeit absolute Orientierungs- und Positionsinformationen herangezogen. Ein Kompass in Verbindung mit Beschleunigungssensoren zur Messung der Gravitation liefert ständig absolute Orientierungsdaten, ein DGPS-Empfänger stellt die notwendigen Positionsdaten zur Verfügung. Ein Kalman-Filter unterstützt die Fusion der Lagedaten dieser unterschiedlichen Verfahren.

Die ersten Arbeiten zum Kalman-Filter wurden in (Kalman, 1960) veröffentlicht. Bekannt wurde es durch seinen erfolgreichen Einsatz in der Raumfahrt. Im Apollo-Raumfahrtprogramm wurde das Kalman-Filter für die Schätzung und Stabilisierung von Trajektorien benutzt, insbesondere bei der Mondlandefähre Apollo (White, 2002). Seither wurde es zu dem am weitesten verbreiteten Schätzer sowohl für lineare als auch für nicht-lineare Systeme.

Der interne Zustand eines Systems oder dessen Veränderung auf Grund von Aktionen kann häufig nicht direkt beobachtet werden. Stattdessen stehen Sensordaten zur Verfügung, die Rückschlüsse auf den gesuchten Systemzustand erlauben. Das Kalman-Filter ist ein rekursiver Datenverarbeitungsalgorithmus, der für die Schätzung dieser internen Systemzustände benutzt werden kann. Die typische Anwendung des Kalman-Filters ist in Abbildung 3.5 dargestellt. Die auf das System wirkenden externen Kräfte und internen Störgrößen werden vom Kalman-Filter verarbeitet. Grundlagen der Schätzung sind:

- ein Modell des zu steuernden Systems und der Messgeräte,

- eine statistische Beschreibung des Systemrauschens, des Messrauschens und der Unsicherheiten in den dynamischen Modellen,
- Informationen über die Anfangswerte der zu bestimmenden Variablen.

Liegt ein lineares Systemmodell vor und kann das System- sowie das Messrauschen durch weißes, gaußsches Rauschen beschrieben werden, so liefert das Kalman-Filter statistisch optimale Ergebnisse. Da der Filteralgorithmus rekursiv arbeitet, ist zur Integration neuer Daten kein Neubeginn der Berechnungen notwendig, vielmehr gehen die neu eintreffenden Daten und die alte Schätzung in gewichteter Form in die Schätzung des neuen Zustands ein.

Liegt kein lineares Systemmodell vor, so muss dieses entweder linearisiert werden, wie beispielsweise beim Einsatz eines INS üblich (Wendel, 2003), oder, wie oft bei der Odometrie radgetriebener mobiler Roboter der Fall, das erweiterte Kalman-Filter angewandt werden (Martinelli und Siegwart, 2003). Letzteres linearisiert das Systemmodell um den aktuellen Zustand.

3.5.1 Formulierung des diskreten Kalman-Filters

Der aktuelle Zustand \mathbf{x}_k eines diskreten, zeitvarianten Systems zum Zeitpunkt k wird durch Gleichung 3.16 (das *Systemmodell*), die Messung \mathbf{z}_k zum Zeitpunkt k durch Gleichung 3.17 (das *Messmodell*) beschrieben (Welch und Bishop, 2001):

$$\mathbf{x}_k = \mathbf{E}_{k-1}\mathbf{u}_{k-1} + \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{s}_{k-1} \quad (3.16)$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{m}_k \quad (3.17)$$

Hierbei repräsentieren \mathbf{s}_k und \mathbf{m}_k das System- bzw. das Messrauschen. Sie werden als voneinander unabhängiges, weißes normalverteiltes Rauschen angenommen:

$$p(\mathbf{m}_k) \sim N(0, \mathbf{M}_k), \quad p(\mathbf{s}_k) \sim N(0, \mathbf{N}_k) \quad (3.18)$$

Matrix \mathbf{F}_{k-1} in Gleichung 3.16 setzt den Zustand des vergangenen Zeitschritts $k-1$ mit dem des aktuellen Zeitschritts k in Verbindung. Matrix \mathbf{E}_{k-1} stellt den Zusammenhang zwischen der optionalen Steuereingabe \mathbf{u}_{k-1} und dem aktuellen Zustand \mathbf{x}_k her. $\mathbf{H}_k\mathbf{x}_k$ in Gleichung 3.17 sagt die Messung \mathbf{z}_k voraus, die auf Grund des Zustands \mathbf{x}_k zu erwarten ist.

Im Folgenden bezeichne $\hat{\mathbf{x}}_k^-$ die a-priori-Zustandsschätzung im Zeitschritt k , d.h. die Zustandsschätzung vor der Integration einer neuen Messung. $\hat{\mathbf{x}}_k$ bezeichne die a-posteriori-Zustandsschätzung zum Zeitpunkt k , also die Zustandsschätzung nachdem die neue Messung berücksichtigt wurde. Die a-priori- und die a-posteriori-Schätzung werden durch Gleichung 3.19 in Beziehung gesetzt:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \quad (3.19)$$

Die Differenz $(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)$ wird *Innovation* genannt. Sie misst die Abweichung der vorhergesagten Messung $\mathbf{H}_k\hat{\mathbf{x}}_k^-$ von der tatsächlichen Messung \mathbf{z}_k . Sind alle Vorbedingungen

(lineares Systemmodell; weißes, gaußsches Messrauschen) erfüllt, ist die Innovation eine Sequenz weißen Rauschens um den Nullpunkt. Diese Eigenschaft kann verwendet werden, um die Optimalität des Filters zu überprüfen und Inkonsistenzen aufzuzeigen. Darüber hinaus können hierdurch Messfehler ausgefiltert werden. Die Matrix K_k , genannt Kalman-Faktor oder Kalman-Gain in Gleichung 3.20, schätzt die Zuverlässigkeit des Systemmodells und des Messmodells und beeinflusst die Korrektur der internen Zustandsschätzung entsprechend:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + M_k)^{-1} \quad . \quad (3.20)$$

In Gleichung 3.20 bezeichnet P_k^- die a-priori-Kovarianzmatrix des Schätzfehlers und M_k die Kovarianzmatrix des Messfehlers. Ist das Messrauschen groß, so geht der Kalman-Faktor K gegen Null. Dem Systemmodell wird mehr Glauben geschenkt. Ist das Systemrauschen groß, so ist auch der Kalman-Faktor K groß. In diesem Fall wird mehr Vertrauen in die Messungen gesetzt.

3.5.2 Kalman-Filter-Algorithmus

Der Kalman-Filter-Algorithmus besteht aus zwei Schritten, einem Vorhersage- oder Prädiktionsschritt (*Time update*) und einem Korrekturschritt (*Data update*). Im Prädiktionsschritt wird mit Hilfe des Systemmodells die a-priori-Schätzung für den nächsten Schritt \hat{x}_k^- aus der Schätzung des aktuellen Zustands \hat{x}_{k-1} propagiert:

$$\hat{x}_k^- = E_{k-1} \mathbf{u}_{k-1} + F_{k-1} \hat{x}_{k-1} \quad . \quad (3.21)$$

Die Prädiktion der Schätzung entspricht nicht dem wahren Zustandsübergang aus Gleichung 3.16; in der Regel sind weder der „echte“ Zustand \mathbf{x}_{k-1} noch der Beitrag des Systemrauschens \mathbf{s}_{k-1} bekannt. Somit muss die Kovarianz des Schätzfehlers ebenfalls fortgeschrieben werden:

$$P_k^- = F_{k-1} P_{k-1} F_{k-1}^T + N_{k-1} \quad . \quad (3.22)$$

Im Korrekturschritt werden neue Messungen integriert, um die a-posteriori-Schätzung zu verbessern. Hierfür wird zuerst der Kalman-Faktor K_k berechnet, der die Kovarianzen des Schätzfehlers und des Messrauschens berücksichtigt:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + M_k)^{-1} \quad . \quad (3.23)$$

Schließlich kann die aktuelle Messung mit der a-priori-Schätzung zur a-posteriori-Schätzung fusioniert werden:

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \quad . \quad (3.24)$$

Die Kovarianz des Schätzfehlers nach der Fusion berechnet sich zu:

$$P_k = (1 - K_k H_k) P_k^- \quad . \quad (3.25)$$

Der Ablauf des Algorithmus ist in Abbildung 3.6 zusammenfassend dargestellt. Der Prädiktionsschritt und der Korrekturschritt müssen sich nicht streng abwechseln. Es können beliebig viele Prädiktionsschritte zwischen zwei Korrekturschritten liegen.

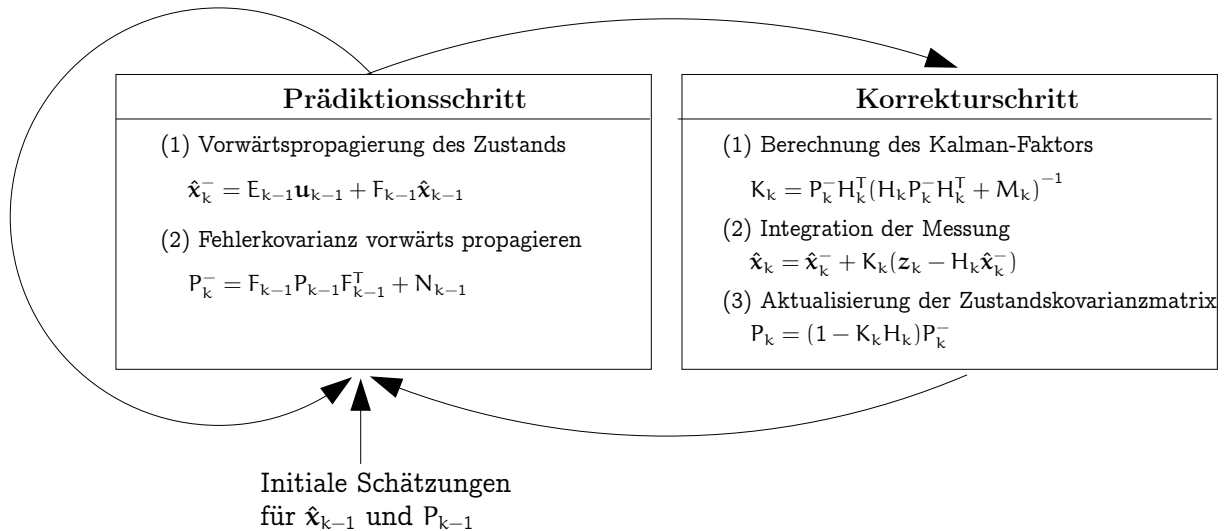


Abbildung 3.6: Ablauf des Kalman-Filters. N und M bezeichnen das System- und das Messrauschen aus Gleichung 3.18 respektive.

3.6 Implementierungsaspekte des Kalman-Filters

Dieser Abschnitt stellt zwei bedeutende Implementierungsaspekte von Kalman-Filtern vor. Einerseits wird zwischen der *direkten* und der *indirekten* Kalman-Filter-Formulierung unterschieden. Andererseits kann das Kalman-Filter mit und ohne Rückkopplung formuliert werden. Die Abgrenzung der Formulierungen wird anhand eines Kalman-Filters für ein Inertialsystem (INS) in Kombination mit einem GPS-Empfänger vorgenommen (Maybeck, 1979). Da auch in dieser Arbeit ein GPS-Empfänger zum Einsatz kommt, wird des Weiteren kurz auf den Aspekt der *schwachen* bzw. *starken* GPS-Kopplung eingegangen.

In den folgenden Darstellungen seien die Position, die Geschwindigkeit und die Orientierung die vom Gesamtsystem (Kalman-Filter in Kombination mit INS und GPS) zu schätzenden Zustände.

3.6.1 Direkte und indirekte Kalman-Filter-Formulierung

Die *direkte* (total-state) Formulierung des Kalman-Filters (Abbildung 3.7) stellt die ursprüngliche Form des Filters dar. Auf diese Formulierung gehen auch die alternativen Bezeichnungen *Time Update* für den Prädiktionsschritt sowie *Data Update* für den Korrekturschritt zurück. Neben der Position, der Geschwindigkeit und der Orientierung schätzt das Kalman-Filter auch die zu messenden Größen, die Beschleunigungen und die Winkelgeschwindigkeiten. Im Filter-Modell werden diese als konstant angenommen. Der Prädiktionsschritt wird ausschließlich durch Voranschreiten der Zeit angestoßen (deshalb auch die Bezeichnung *Time Update*). Der aktuell geschätzte Systemzustand und die Fehlerkovarianz werden mit Hilfe des Systemmodells vorwärtspropagiert (vergleiche auch Abbildung 3.6), also die zeitliche Integration der Beschleunigungen, Winkelgeschwindigkeiten und Geschwindigkeiten durchgeführt. Treffen neue IMU- oder GPS-

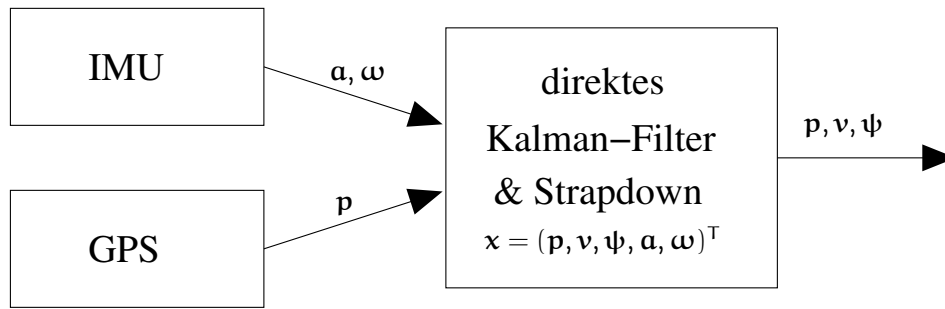


Abbildung 3.7: Das direkte Kalman-Filter mit geschätztem Systemzustand \mathbf{x} (Position \mathbf{p} , Geschwindigkeit \mathbf{v} , Orientierung $\boldsymbol{\psi}$, Beschleunigung \mathbf{a} und Rotationsgeschwindigkeit $\boldsymbol{\omega}$). Sensormessungen werden von einer IMU (\mathbf{a} , $\boldsymbol{\omega}$, siehe hierzu auch Abbildung 3.2) und von einem GPS-System (\mathbf{p}) geliefert.

Messungen ein, wird ein Korrekturschritt initiiert (deshalb auch die Bezeichnung *Data Update*). Die Messungen werden mit dem aktuellen Systemzustand „statistisch optimal“ verrechnet und die Schätzungen des Systemzustands korrigiert sowie die Zustandskovarianzen aktualisiert.

In dieser Konfiguration ist das Kalman-Filter somit fester Bestandteil des INS. Dies wirkt sich in zweierlei Hinsicht nachteilig aus. Zum einen müssen die Bewegungen des Roboters mit einer relativ hohen Frequenz verfolgt, Fehler unterdrückt und das Kalman-Filter fortgeschrieben werden. Zum anderen arbeitet der Navigationsalgorithmus nicht, sollte das Kalman-Filter ausfallen.

Wird eine *indirekte* (error-state) Kalman-Filter-Formulierung gewählt, so schätzt das Filter nicht mehr direkt die absoluten Größen Position, Geschwindigkeit und Orientierung, sondern deren Fehler in den vom INS berechneten Werten. Dem Filter wird die Differenz zwischen der INS- und der GPS-Position als Messung präsentiert. Jede Messung veranlasst einen Korrekturschritt, durch den die Fehler des INS neu geschätzt werden. Im Prädiktionsschritt werden die geschätzten Fehler durch das Modell zeitlich vorwärtspropagiert.

Da sich das indirekte Kalman-Filter außerhalb des INS-Zyklus befindet, kann seine Abtastrate wesentlich niedriger sein als die des direkten Filters. Der Berechnungsaufwand wird damit wesentlich geringer. Ein weiterer Vorteil des indirekten Kalman-Filters besteht darin, dass der INS-Navigationsalgorithmus weiterarbeiten kann, auch wenn das Kalman-Filter ausfällt.

3.6.2 Formulierung mit und ohne Rückkopplung

Das indirekte Kalman-Filter kann sowohl mit (*feedback*) als auch ohne (*feedforward*) Rückkopplung realisiert werden. Im Falle der *feedforward*-Formulierung wird die Fehlerschätzung benutzt, um die aktuelle Positions-, Geschwindigkeits- und Orientierungsschätzung zu korrigieren, ohne jedoch das INS zu aktualisieren (Abbildung 3.8). Das INS arbeitet unabhängig vom korrigierenden Filter. Da die Korrekturen das INS nicht direkt beeinflussen, besteht jedoch das Problem, dass die vom INS errechnete Position mit un-

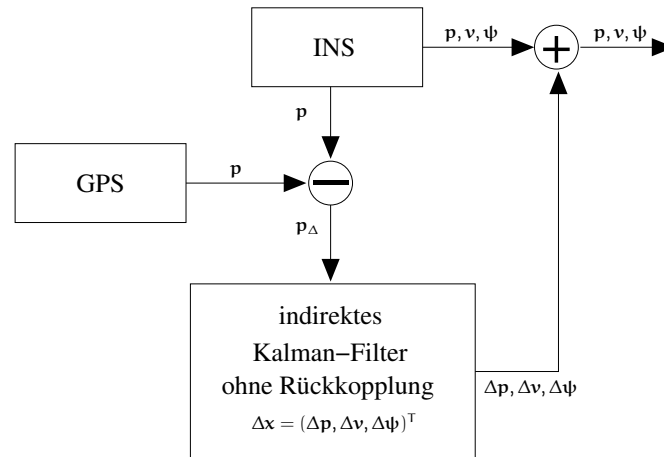


Abbildung 3.8: Das indirekte Kalman-Filter ohne Rückkopplung mit geschätztem Systemzustand $\Delta \mathbf{x}$ (Positions- $\Delta \mathbf{p}$, Geschwindigkeits- $\Delta \mathbf{v}$ und Orientierungsfehler $\Delta \psi$). Als Sensormessung wird dem Kalman-Filter die Differenz p_{Δ} zwischen INS- und GPS-Position präsentiert.

begrenztem Fehler driften kann. Im Falle der *feedback*-Formulierung (Abbildung 3.9) kann das indirekte Kalman-Filter darüber hinaus Schätzungen für die inhärenten Fehler des Inertialsystems, beispielsweise die Biase und Skalierungsfehler, berechnen. Alle geschätzten Fehler werden zum INS zurückgeführt, um dort neben der Position, der Geschwindigkeit und der Orientierung auch noch die Beschleunigungsmesswerte und die Drehraten direkt zu korrigieren. Nach der Rückführung der Fehler zum INS muss hierbei beachtet werden, dass der Systemzustand, also die geschätzten Fehler, nach jedem Korrekturschritt auf Null zurückgesetzt werden müssen. Dem INS dürfen die errechneten Korrekturen nicht mehrfach präsentiert werden.

Dies bringt wiederum einige Vorteile mit sich. Zum einen können dadurch die Fehler des Inertialsystems nicht mehr unbegrenzt wachsen. Zum anderen führt die weitere Verwendung von korrigierten Beschleunigungs- und Drehratenmessungen im INS zur dauerhaften Verbesserung des Systems (Maybeck, 1979).

3.6.3 Schwach und stark gekoppelte INS-GPS-Integration

Bei der Stützung eines Inertialsystems mit einem GPS-System mittels eines indirekten Kalman-Filters wird weiterhin zwischen schwach gekoppelten (Loosely-Coupled) und stark gekoppelten (Tightly-Coupled) Systemen unterschieden. Ein schwach gekoppeltes System liegt vor, wenn die Korrektur des INS direkt durch GPS Positions- und Geschwindigkeitsmessungen erfolgt. Der zur Integration der Sensoren nötige Aufwand ist hierbei am geringsten.

Ein stark gekoppeltes System hingegen verarbeitet GPS-Rohdaten, wie Pseudoentfernungen oder satellitenbezogene Daten im Kalman-Filter (Wendel, 2003). Seine Realisierung ist im Vergleich zu schwach gekoppelten Systemen wesentlich aufwändiger.

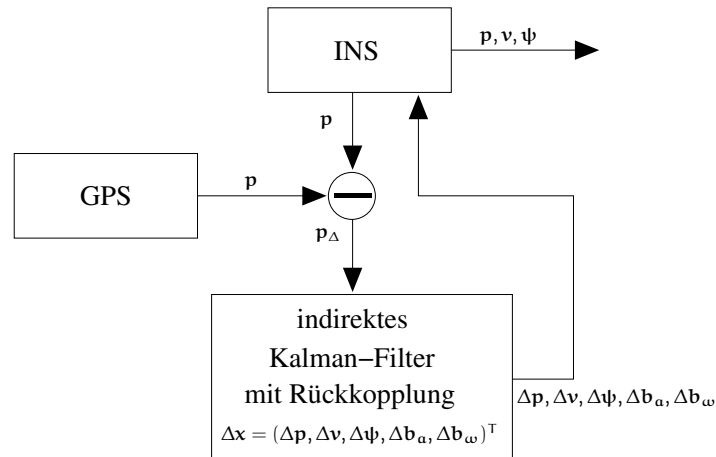


Abbildung 3.9: Das indirekte Kalman-Filter mit Rückkopplung und geschätztem Systemzustand $\Delta \mathbf{x}$ (Positions- $\Delta \mathbf{p}$, Geschwindigkeits- $\Delta \mathbf{v}$, Orientierungsfehler $\Delta \boldsymbol{\psi}$ sowie Beschleunigungsbias $\Delta \mathbf{b}_a$ und Gyroskopbias $\Delta \mathbf{b}_\omega$). Die Differenz \mathbf{p}_Δ zwischen INS- und GPS-Position dient als Sensormessung des Kalman-Filter.

3.7 Das realisierte Kalman-Filter

Um ein Kalman-Filter realisieren zu können, muss ein Modell des zu beobachtenden Systems vorhanden sein. Die Grundlage des in dieser Arbeit zu entwerfenden Kalman-Filters bildet die Odometrie von Laufmaschinen (vergleiche Abschnitt 3.4). Im Gegensatz zur Odometrie radgetriebener Systeme scheint es jedoch nur schwer möglich, das Systemmodell für das Kalman-Filter auf Basis der komplexen Berechnungen aus Abschnitt 3.4 aufzustellen. Darüber hinaus macht es nach Abschnitt 3.2.1.1 wenig Sinn, systematische Fehler in der Berechnung zu modellieren, da im Gelände nicht-systematische Einflüsse dominieren.

Ogleich keine systematischen Fehler mitgeschätzt werden, bringt die indirekte, rückgekoppelte Formulierung des Kalman-Filters dennoch einen entscheidenden Vorteil mit sich: Nichtlinearitäten bei der Berechnung von Orientierungskorrekturen werden geschickt umgangen, da der Fehlerzustandsvektor des Filters nach jedem Korrekturschritt wieder auf Null gesetzt wird (vergleiche Abschnitt 3.6.2). Auf die Realisierung eines stark gekoppelten Systems zur GPS-Integration wird auf Grund des immensen Aufwands jedoch verzichtet.

Im Folgenden wird zur Fusion der Lokalisationsinformationen von Odometrie, Kompass & Gravitation, sowie GPS ein schwach gekoppeltes, indirektes Kalman-Filter mit Rückkopplung wie in Abbildung 3.10 skizziert entwickelt und hergeleitet.

3.7.1 Fortschreibung der Lage

Zur Modellierung steht ausschließlich das Ergebnis der Odometrieberechnung zur Verfügung, bestehend aus Position \mathbf{p}^s und Orientierung $\boldsymbol{\psi}^s$ nach Gleichung 3.2. Um dem relativ messenden Charakter der Odometrie Rechnung zu tragen, sollen dem Kalman-Filter relative Messungen im Roboterkoordinatensystem in Form von *Geschwindigkeit*

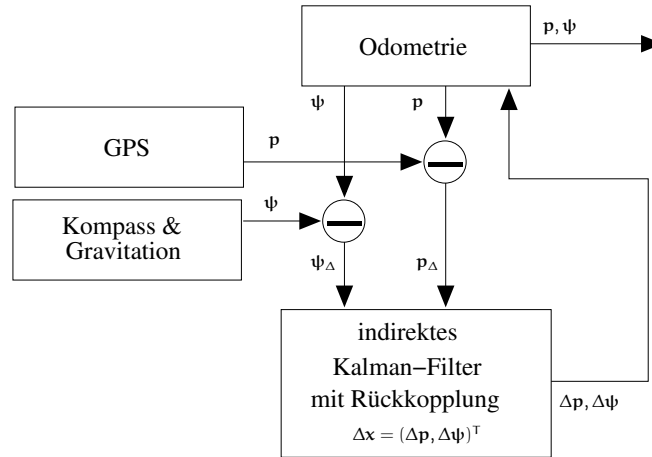


Abbildung 3.10: Das realisierte indirekte Kalman-Filter mit Rückkopplung und geschätztem Systemzustand Δx (Positions- Δp und Orientierungsfehler $\Delta \psi$). Die Differenzen p_Δ zwischen Odometrie- und GPS-Position sowie ψ_Δ zwischen Odometrieorientierung und Orientierungssensor dienen als Sensormessungen des Kalman-Filter.

ten v^r und Winkelgeschwindigkeiten ω^r präsentiert werden. Hierzu muss die von der Odometrie berechnete Lage zunächst differenziert werden.

In diskreter Form mit Zeitindex k ist die Rotationsmatrix $R_{\Delta,k}^r$ gesucht, welche die Orientierungsänderung des Roboters $\psi_{\Delta,k}^r$ im Zeitschritt k beschreibt. Ausgehend von der Roboterorientierung ψ_k^g zum Zeitpunkt k , repräsentiert durch die Rotationsmatrix $R_{r,k}^g$, gilt:

$$\begin{aligned} R_{r,k}^g &= R_{r,k-1}^g R_{\Delta,k}^r \\ R_{\Delta,k}^r &= R_{r,k-1}^{gT} R_{r,k}^g \end{aligned} \quad (3.26)$$

Um die Orientierungsänderung des Roboters $\psi_{\Delta,k}^r$ zu bestimmen, muss die Rotationsmatrix des letzten Zeitschritts $k-1$ transponiert und von links an die Rotationsmatrix des Zeitschritts k multipliziert werden. Unter der Annahme einer gleichmäßigen Rotation während des Abtastintervalls berechnen sich die benötigten Winkelgeschwindigkeiten in Roboterkoordinaten zu:

$$\omega_k^r = \frac{\psi_{\Delta,k}^r}{t_\Delta} \quad (3.27)$$

Die gesuchten Geschwindigkeiten im Roboterkoordinatensystem ergeben sich aus folgendem Zusammenhang:

$$v_k^r = \frac{R_{r,k-1}^{gT} (p_k^g - p_{k-1}^g)}{t_\Delta} \quad (3.28)$$

Hierbei wird die Richtung der Geschwindigkeiten auf den Zeitschritt $k-1$ bezogen.

Der 6-dimensionale Vektor $(v^r, \omega^r)^T$ bildet schließlich die Grundlage für die weiteren Berechnungen im Kalman-Filter. Theoretisch muss mit diesen Daten, analog dem Strapdown-Algorithmus zur Lageberechnung im INS (vergleiche Abschnitt 3.2.1.2), die

Lage des Roboters ständig fortgeschrieben werden. In der Praxis kann jedoch auf die explizite Lagefortschreibung auf Basis der berechneten Werte $(\mathbf{v}^r, \boldsymbol{\omega}^r)^\top$ verzichtet werden; diese wird eben schon von der Odometrieberechnung geleistet. Trotzdem sind die folgenden Gleichungen zur „virtuellen“ Fortschreibung der Lage für die Modellierung des Filters erforderlich. Die neue Position ergibt sich durch Addition der rotierten, integrierten Geschwindigkeiten:

$$\mathbf{p}_k^g = \mathbf{p}_{k-1}^g + \mathbf{R}_{r,k-1}^g \mathbf{v}_k^r t_\Delta \quad . \quad (3.29)$$

Die Integration der Winkelgeschwindigkeiten ergibt die Orientierungsänderung

$$\boldsymbol{\psi}_{\Delta,k}^r = \boldsymbol{\omega}_k^r t_\Delta \quad , \quad (3.30)$$

mit Hilfe deren Rotationsmatrix $\mathbf{R}_{\Delta,k}^r$ berechnet sich die neue Orientierung:

$$\mathbf{R}_{r,k}^g = \mathbf{R}_{r,k-1}^g \mathbf{R}_{\Delta,k}^r \quad . \quad (3.31)$$

3.7.2 Das nichtlineare Kalman-Filter-Modell

Ausgangspunkt für die an Wendel (2003) angelehnte Herleitung des Kalman-Filters ist das nichtlineare Systemmodell (die Notation $\dot{\mathbf{x}}$ bezeichne im Folgenden die zeitliche Ableitung des Vektors \mathbf{x}):

$$\dot{\mathbf{x}} = f(\mathbf{x}) \quad (3.32)$$

mit Zustandsvariable $\mathbf{x} = (\mathbf{p}^g, \boldsymbol{\psi}^g)^\top$. Die rechte Seite wird um den Punkt $\hat{\mathbf{x}}$ in eine Taylor-Reihe entwickelt, die nach dem linearen Glied abgebrochen wird (vergleiche hierzu die in Anhang A.4 beschriebene mehrdimensionale Taylor-Entwicklung). $\hat{\mathbf{x}}$ bezeichnet hierbei den geschätzten aktuellen Systemzustand:

$$\dot{\mathbf{x}} \approx f(\hat{\mathbf{x}}) + \left. \frac{\delta f(\mathbf{x})}{\delta \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}} (\mathbf{x} - \hat{\mathbf{x}}) \quad . \quad (3.33)$$

Mit $\dot{\hat{\mathbf{x}}} = f(\hat{\mathbf{x}})$, $\Delta \mathbf{x} = \mathbf{x} - \hat{\mathbf{x}}$ und $\mathbf{F} = \left. \frac{\delta f(\mathbf{x})}{\delta \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}}$ erhält das gesuchte Systemmodell die Form

$$\Delta \dot{\mathbf{x}} = \mathbf{F} \Delta \mathbf{x} \quad . \quad (3.34)$$

Das heißt, zur Abschätzung des Fehlers wird die Taylor-Entwicklung aus Gleichung 3.33 umformuliert. Folgende Fehler werden durch das Kalman-Filter geschätzt:

- der Positionsfehlervektor $\Delta \mathbf{p}^g = (\Delta x, \Delta y, \Delta z)^\top$ im globalen Koordinatensystem
- der Orientierungsfehlervektor $\Delta \boldsymbol{\psi}^g = (\Delta \alpha, \Delta \beta, \Delta \gamma)^\top$ bezüglich des globalen Koordinatensystems; dieser ist auch repräsentiert durch die Rotationsmatrix \mathbf{R}_{Δ}^g

Der Zustandsvektor des gesuchten indirekten Kalman-Filters ergibt sich, indem man obige Fehlervektoren in dem 6-komponentigen Fehlervektor

$$\Delta \mathbf{x} = (\Delta x, \Delta y, \Delta z, \Delta \alpha, \Delta \beta, \Delta \gamma)^\top \quad (3.35)$$

zusammenfasst.

Nun wird Gleichung 3.42 in Gleichung 3.36 eingesetzt und ausmultipliziert:

$$\begin{aligned}\dot{\mathbf{p}}^g &= (1 + \Psi_\Delta^g) \hat{\mathbf{R}}_r^g \mathbf{v}^r \\ &= \hat{\mathbf{R}}_r^g \mathbf{v}^r + \Psi_\Delta^g \hat{\mathbf{R}}_r^g \mathbf{v}^r \quad .\end{aligned}\quad (3.43)$$

Nutzt man die Eigenschaft

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}] \mathbf{b} = -[\mathbf{b}] \mathbf{a} = -\mathbf{b} \times \mathbf{a} \quad (3.44)$$

des Kreuzprodukts, wobei

$$[\mathbf{a}] = \begin{bmatrix} \left(\begin{array}{c} a_x \\ a_y \\ a_z \end{array} \right) \end{bmatrix} = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix} \quad (3.45)$$

eine so genannte schief-symmetrische Matrix darstellt, wird Gleichung 3.43 zu

$$\begin{aligned}\dot{\mathbf{p}}^g &= \hat{\mathbf{R}}_r^g \mathbf{v}^r + [\Delta\psi^g] \hat{\mathbf{R}}_r^g \mathbf{v}^r \\ &= \hat{\mathbf{R}}_r^g \mathbf{v}^r - [\hat{\mathbf{R}}_r^g \mathbf{v}^r] \Delta\psi^g \quad .\end{aligned}\quad (3.46)$$

Diese Differentialgleichung ist Grundlage für die Berechnungen nach den Gleichungen 3.33 und 3.34. Bei der durch die Taylor-Entwicklung notwendigen Ableitung ergeben sich nur für ψ^g von Null verschiedene Werte, so dass die Differentialgleichung des Positionsfehlers schließlich folgendermaßen formuliert werden kann:

$$\Delta\dot{\mathbf{p}}^g = -[\hat{\mathbf{R}}_r^g \mathbf{v}^r] \Delta\psi^g \quad . \quad (3.47)$$

3.7.2.2 Herleitung der Orientierungsfehlerdifferentialgleichung

Für den Orientierungsfehler im Roboterkoordinatensystem kann nach Gleichung 3.30 folgende Differentialgleichung aufgestellt werden:

$$\dot{\boldsymbol{\psi}}^r = \boldsymbol{\omega}^r \quad . \quad (3.48)$$

Für den Orientierungsfehler in globalen Koordinaten gilt:

$$\dot{\boldsymbol{\psi}}^g = \mathbf{R}_r^g \boldsymbol{\omega}^r \quad . \quad (3.49)$$

Mit den Ausführungen des letzten Abschnittes erhält man schließlich entsprechend der Positionsdifferentialgleichung 3.46 die Orientierungsdifferentialgleichung:

$$\dot{\boldsymbol{\psi}}^g = \mathbf{R}_r^g \boldsymbol{\omega}^r - [\hat{\mathbf{R}}_r^g \boldsymbol{\omega}^r] \Delta\psi^g \quad . \quad (3.50)$$

Die Taylor-Entwicklung führt schließlich zur Differentialgleichung des Orientierungsfehlers:

$$\Delta\dot{\boldsymbol{\psi}}^g = -[\hat{\mathbf{R}}_r^g \boldsymbol{\omega}^r] \Delta\psi^g \quad . \quad (3.51)$$

3.7.2.3 Zusammenfassung in einer Fehlerdifferentialgleichung

Werden die Gleichungen 3.47 und 3.51 in Form einer Matrixengleichung dargestellt und das durch die Matrix G in den Zustandsraum transformierte Systemrauschen s addiert, so erhält man

$$\underbrace{\begin{pmatrix} \Delta \mathbf{p} \\ \Delta \boldsymbol{\psi} \end{pmatrix}}_{\Delta \dot{\mathbf{x}}} = \underbrace{\begin{pmatrix} 0 & -[\hat{\mathbf{R}}_r^g \mathbf{v}^r] \\ 0 & -[\hat{\mathbf{R}}_r^g \boldsymbol{\omega}^r] \end{pmatrix}}_F \underbrace{\begin{pmatrix} \Delta \mathbf{p} \\ \Delta \boldsymbol{\psi} \end{pmatrix}}_{\Delta \mathbf{x}} + \underbrace{\begin{pmatrix} \mathbf{R}_r^g & 0 \\ 0 & \mathbf{R}_r^g \end{pmatrix}}_G \underbrace{\begin{pmatrix} \mathbf{s}_{\mathbf{v}^r} \\ \mathbf{s}_{\boldsymbol{\omega}^r} \end{pmatrix}}_s . \quad (3.52)$$

3.7.3 Diskretes Kalman-Filter-Modell

Die Lösung der Differentialgleichung

$$\Delta \dot{\mathbf{x}} = F \Delta \mathbf{x} \quad (3.53)$$

lässt sich durch

$$\Delta \mathbf{x}(t) = \Phi(t) \Delta \mathbf{x}(0) \quad (3.54)$$

ausdrücken, mit $\Phi(0) = 1$ und $\dot{\Phi}(t) = F \Phi(t)$. Betrachtet man nur Diskretisierungszeitpunkte $t = kt_\Delta$ und nimmt an, dass F innerhalb eines Zeitintervalls t_Δ konstant ist, so berechnet sich Φ_k zu:

$$\Phi_k = e^{F_k t_\Delta} . \quad (3.55)$$

Wird die Exponentialfunktion durch die ersten zwei Glieder ihrer Reihenentwicklung approximiert ($\Phi_k \approx 1 + F_k t_\Delta$), ergibt sich als Diskretisierung der Gleichung 3.53

$$\begin{aligned} \Delta \mathbf{x}_{k+1} &= \Phi_k \Delta \mathbf{x}_k \\ &= (1 + F_k t_\Delta) \Delta \mathbf{x}_k . \end{aligned} \quad (3.56)$$

Die Matrix für das Systemrauschen N_k berechnet sich nach Gleichung 3.57, welche beispielsweise in (Wendel, 2003) hergeleitet wird:

$$N_k = \Phi_k G_k N G_k^T \Phi_k^T . \quad (3.57)$$

N entspricht einer 6×6 Kovarianzmatrix des Geschwindigkeits- und Winkelgeschwindigkeitsrauschens.

Gleichung 3.57 gilt nur, wenn der Abstand t_Δ zwischen zwei Orientierungs- bzw. Positionsmessungen konstant ist. Sonst entstehen durch obige Modellierung des Rauschens kleine Fehler. Das Rauschen für variable Zeitabstände $t_{\Delta,k}$ berechnet sich durch Ersetzen von N durch die spektrale Leistungsdichte N_t multipliziert mit der verstrichenen Zeit $t_{\Delta,k}$ durch:

$$N_k = \Phi_k G_k N_t t_{\Delta,k} G_k^T \Phi_k^T . \quad (3.58)$$

3.7.4 Besondere Eigenschaften des entwickelten Systemmodells

Wird das Systemmodell F mit Modellen für ähnliche Anwendungen verglichen, so zeichnet es sich vor allem durch seine flexible, allgemeine Anwendbarkeit aus. Das Modell macht keine Annahmen über die Roboterdynamik. Der Roboter kann sich in der xy -Ebene bewegen. Er kann jedoch auch an einer Wand emporklettern. Beide Bewegungen werden vom Modell abgedeckt. In den bisherigen Anwendungen des indirekten Kalman-Filters bei mobilen Robotern wurden zumeist nur zweidimensionale Modelle verwendet.

Des Weiteren gleicht sich das Systemmodell F dynamisch an aktuelle Gegebenheiten an. Bei hohen Geschwindigkeiten bzw. Winkelgeschwindigkeiten wirken sich Winkelfehler stärker aus. Diese Komponente fließt über die schief-symmetrische Matrix $[\hat{R}_r^g \mathbf{v}^r]$ bzw. $[\hat{R}_r^g \boldsymbol{\omega}^r]$ in das Systemmodell F ein. Bei hohen Geschwindigkeiten modelliert das Systemmodell eine höhere Kovarianz zwischen den Winkel- und Geschwindigkeitsfehlern; bei hohen Winkelgeschwindigkeiten eine höhere Winkelfehlerkovarianz. Dies wirkt sich wiederum auf die Zustandskovarianzmatrix P aus.

3.8 Ablauf des Lokalisationsalgorithmus

Nachdem nun alle nötigen Modelle hergeleitet und beschrieben sind, fasst dieser Abschnitt das vorliegende Lokalisationskonzept anhand der Teilschritte des Lokalisationsalgorithmus zusammen. Abbildung 3.11 veranschaulicht die folgenden Ausführungen in graphischer Form.

3.8.1 Odometrie

Vor der eigentlichen Odometrieberechnung werden aus den zur Verfügung stehenden Sensordaten (Fußkräfte und Beinbewegungen) mit Hilfe von Fuzzy-Bewertern die Beinbewegungen g_j bestimmt. Anschließend wird die neue Lage des Roboters $(\hat{\mathbf{p}}_k^g, \hat{\boldsymbol{\psi}}_k^g)^T$ nach Gleichung 3.12 geschätzt. Nach Abschnitt 3.7.1 stehen hierdurch die für das Kalman-Filter notwendigen Eingangswerte $(\mathbf{v}_k^r, \boldsymbol{\omega}_k^r)^T$ im Roboterkoordinatensystem zu Verfügung.

3.8.2 Prädiktionsschritt

Im Kalman-Filter lösen die neu eintreffenden Odometriedaten einen Prädiktionsschritt aus. Eine Steuereingabe u wurde im vorliegenden Kalman-Filter nicht modelliert, so dass das Vorwärtspropagieren des Zustandes nach Gleichung 3.21 folgende Form annimmt:

$$\Delta \mathbf{x}_k = \Phi_k \Delta \mathbf{x}_{k-1} \quad . \quad (3.59)$$

Zu beachten ist hierbei, dass das Systemmodell in jedem Prädiktionsschritt neu berechnet werden muss, da es nach Gleichung 3.52 direkt von den Eingangswerten $(\mathbf{v}_k^r, \boldsymbol{\omega}_k^r)^T$ abhängig ist.

Bei einer Formulierung als rückgekoppeltes indirektes Kalman-Filter wird angenommen, dass nach einem Korrekturschritt alle Fehler korrigiert sind und deshalb der Fehlerzustandsvektor auf Null zurückgesetzt werden muss ($\Delta \mathbf{x}_{k-1} = \mathbf{0}$). Berücksichtigt man

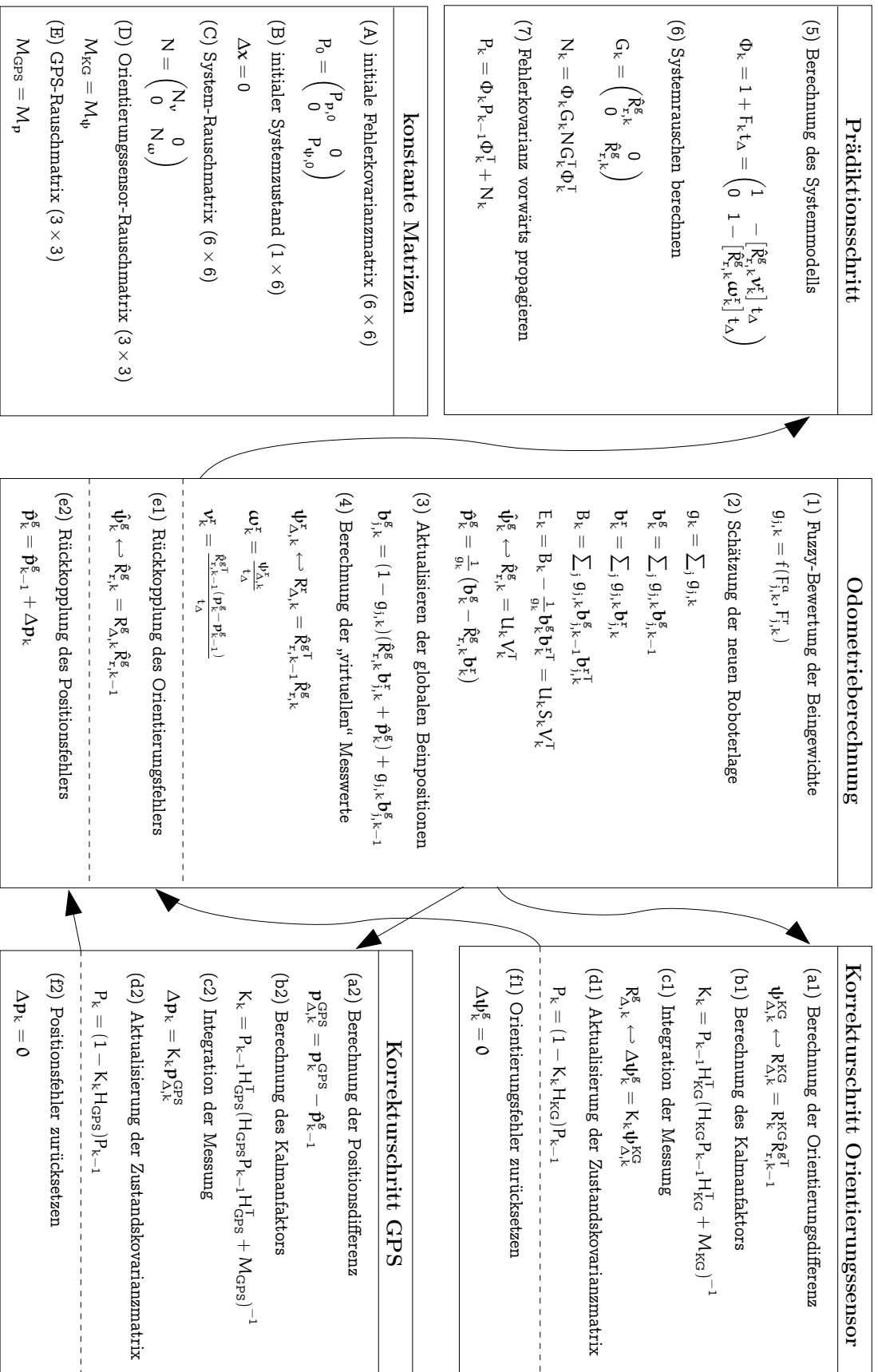


Abbildung 3.11: Zusammenfassende Darstellung des Lokalisationsalgorithmus: in der Mitte die Odometrieberechnung, links der Prädiktionsschritt des Kalman-Filters, rechts die Orientierungs- und Positionskorrekturen durch Kompass & Gravitation bzw. GPS.

dies, so gilt nach Gleichung 3.59 $\Delta \mathbf{x}_k = \mathbf{0}$. Somit kann auf das Vorwärtspropagieren des Zustandes gänzlich verzichtet werden. Im nächsten Schritt wird das Systemrauschen \mathbf{N}_k nach Gleichung 3.57 berechnet und anschließend die Kovarianz der Schätzung entsprechend Gleichung 3.22 vorwärtspropagiert:

$$\mathbf{P}_k = \Phi_k \mathbf{P}_{k-1} \Phi_k^T + \mathbf{N}_k \quad . \quad (3.60)$$

3.8.3 Positionskorrektur durch GPS

Beim Eintreffen einer neuen gültigen GPS-Position $\mathbf{p}_k^{\text{GPS}}$ kann dem Kalman-Filter ein neuer Messfehler präsentiert werden:

$$\mathbf{p}_{\Delta,k}^{\text{GPS}} = \mathbf{p}_k^{\text{GPS}} - \hat{\mathbf{p}}_{k-1}^g \quad . \quad (3.61)$$

Da der Messfehlervektor $\mathbf{p}_{\Delta,k}^{\text{GPS}}$ aus drei Komponenten besteht und der Fehlerzustandsvektor sechs Komponenten aufweist, ergibt sich folgende 3×6 Matrix

$$\mathbf{H}_{\text{GPS}} = \begin{pmatrix} 1 & 0 \end{pmatrix} \quad , \quad (3.62)$$

um die GPS-Daten mit dem Zustandsvektor in Verbindung zu setzen. Mit der 3×3 Kovarianzmatrix \mathbf{M}_{GPS} des GPS-Positionsrauschens berechnet sich der Kalman-Faktor dann nach Gleichung 3.23 zu:

$$\mathbf{K}_k^{\text{GPS}} = \mathbf{P}_{k-1} \mathbf{H}_{\text{GPS}}^T (\mathbf{H}_{\text{GPS}} \mathbf{P}_{k-1} \mathbf{H}_{\text{GPS}}^T + \mathbf{M}_{\text{GPS}})^{-1} \quad . \quad (3.63)$$

Berücksichtigt man wieder $\Delta \mathbf{p}_{k-1} = \mathbf{0}$, so wird die Integration der Messung nach Gleichung 3.24 zu:

$$\Delta \mathbf{p}_k = \mathbf{K}_k^{\text{GPS}} \mathbf{p}_{\Delta,k}^{\text{GPS}} \quad . \quad (3.64)$$

Schließlich muss die Kovarianzmatrix der Schätzung entsprechend Gleichung 3.25 aktualisiert werden:

$$\mathbf{P}_k = (\mathbf{1} - \mathbf{K}_k^{\text{GPS}} \mathbf{H}_{\text{GPS}}) \mathbf{P}_{k-1} \quad . \quad (3.65)$$

Nun werden die berechneten Korrekturen rückgekoppelt und zur geschätzten Lage addiert. Für die Position kann folgende Gleichung angegeben werden:

$$\hat{\mathbf{p}}_k^g = \hat{\mathbf{p}}_{k-1}^g + \Delta \mathbf{p}_k \quad . \quad (3.66)$$

Zuletzt muss der Fehlerzustandsvektor noch auf Null zurückgesetzt werden:

$$\Delta \mathbf{p}_k = \mathbf{0} \quad . \quad (3.67)$$

3.8.4 Orientierungskorrektur durch Kompass & Gravitation

Trifft ein neues Datum ψ_k^{KG} (korrespondierende Rotationsmatrix R_k^{KG}) vom durch Kompass & Gravitation realisierten absoluten Orientierungsmesser ein, kann ebenfalls ein Korrekturschritt im Kalman-Filter durchgeführt werden. Der Messfehler $\psi_{\Delta,k}^{\text{KG}}$ kann hierbei nur indirekt über die Rotationsmatrix $R_{\Delta,k}^{\text{KG}}$ bestimmt werden:

$$R_{\Delta,k}^{\text{KG}} = R_k^{\text{KG}} \hat{R}_{r,k-1}^{\text{gT}} \quad (3.68)$$

Die Verbindung zwischen den Orientierungsdaten und dem Zustandsvektor stellt die 3×6 Matrix

$$H_{\text{KG}} = \begin{pmatrix} 0 & 1 \end{pmatrix} \quad (3.69)$$

her. Mit der 3×3 Kovarianzmatrix M_{KG} des Orientierungsmessrauschens berechnet sich nacheinander der Kalman-Faktor und die a-posteriori-Schätzung des Fehlerzustandsvektors:

$$K_k^{\text{KG}} = P_{k-1} H_{\text{KG}}^T (H_{\text{KG}} P_{k-1} H_{\text{KG}}^T + M_{\text{KG}})^{-1} \quad , \quad (3.70)$$

$$\Delta\psi_k^{\text{g}} = K_k^{\text{KG}} \psi_{\Delta,k}^{\text{KG}} \quad . \quad (3.71)$$

Genau an dieser Stelle kommt der Vorteil der indirekten Formulierung des Kalman-Filters zum Tragen. Bei der direkten Formulierung entsprechend Gleichung 3.24 auf Seite 3.24 müssten an dieser Stelle Orientierungen addiert werden, was in dieser Form nicht erlaubt wäre. Anschließend wird noch die a-posteriori-Kovarianz des Schätzfehlers berechnet:

$$P_k = (1 - K_k^{\text{KG}} H_{\text{KG}}) P_{k-1} \quad . \quad (3.72)$$

Zur Rückkopplung der Korrekturen muss bei der Orientierungsberechnung auf die zugehörigen Rotationsmatrizen zurückgegriffen werden, da die direkte Addition von Rotationen nicht zulässig ist. Der Orientierungsfehler $\Delta\psi_k^{\text{g}}$ aus Gleichung 3.71 definiert die Rotationsmatrix $R_{\Delta,k}^{\text{g}}$, mit welcher die aktuelle Schätzung der Rotationsmatrix $\hat{R}_{r,k}^{\text{g}}$ korrigiert wird zu:

$$\hat{R}_{r,k}^{\text{g}} = R_{\Delta,k}^{\text{g}} \hat{R}_{r,k-1}^{\text{g}} \quad . \quad (3.73)$$

Schließlich wird der Fehlerzustandsvektor wieder auf Null zurückgesetzt:

$$\Delta\psi_k^{\text{g}} = 0 \quad . \quad (3.74)$$

3.9 Zusammenfassung

Zur Lösung der Lokalisationsproblematik werden in der Literatur unterschiedliche Ansätze vorgeschlagen. Doch nicht alle sind für den Einsatz auf Laufmaschinen geeignet.

Die Navigation von Laufmaschinen im Gelände stellt spezielle Anforderungen an die Verfahren: Es darf nur gewichtsmäßig leichte, rechner-effiziente Sensorik zum Einsatz kommen. Positionssprünge und unbegrenzt wachsende Fehler gilt es zu vermeiden. Unstrukturiertes Gelände ist vorab selten bekannt und weist keine strukturierten Merkmale auf. Darüber hinaus herrschen oft schwierige Bodenverhältnisse und zeitvariante Umgebungsbedingungen vor. Dem kommt erschwerend hinzu, dass Laufmaschinen mit ihrer hohen Zahl an Bewegungsfreiheitsgraden eine vollständige Lagebeschreibung im Raum erfordern. Sie bewegen sich meist recht langsam ohne durchgängigen Kontakt mit dem Untergrund.

Es zeigt sich, dass eine Kombination aus relativ und absolut arbeitenden Verfahren zur Einhaltung dieser Anforderungen notwendig ist. Inertialsysteme scheitern vor allem an den langsamen, unruhigen Roboterbewegungen, so dass die Odometrie als systemimmanentes, relativ messendes Verfahren gewählt wird. Um die durch rutschende Beine verursachten Odometriefehler zu reduzieren, findet eine Gewichtung der Stemmbeine auf dem kontinuierlichen Wertebereich $[0, 1]$ statt. Die Beingewichte werden mit Methoden der Fuzzy-Logik bestimmt.

Aus dem Bereich der absoluten Verfahren bieten sich Beschleunigungssensoren in Kombination mit Magnetfeldsensoren an, um die absolute Orientierung der Laufmaschine im Raum zu bestimmen. Die Erdbeschleunigung weist nach unten, der Kompass in Richtung Norden. Die Tatsache, dass sich die Laufmaschine in freiem Gelände bewegen soll, erlaubt weiterhin den Einsatz eines DGPS-Empfängers, der bei freier Sicht die absolute Position im Raum liefern kann. Auf den Einsatz von landmarken- und kartenbasierten Verfahren wird hingegen verzichtet. Durch die gestellten Anforderungen steigt die Komplexität dieser Verfahren derart, dass eine echtzeitfähige Lösung für autonome Laufmaschinen derzeit noch unmöglich scheint.

Eine möglichst optimale Methode zur Fusion der unterschiedlichen Lokalisationsinformationen stellt das Kalman-Filter bereit. Das in dieser Arbeit realisierte schwach gekoppelte, indirekte Kalman-Filter mit Rückkopplung macht keinerlei Annahmen hinsichtlich der Roboterdynamik und passt sich dynamisch an aktuelle Gegebenheiten an.

Mit den gewählten Lokalisationsmethoden Odometrie, Kompass & Gravitation sowie GPS beschränkt sich das vorgestellte Lokalisationskonzept auf einfache und rechner-effiziente Sensorik. Den speziellen Anforderungen sowohl an unstrukturiertes Gelände als auch an Laufmaschinen wird hierbei Rechnung getragen. Wie nun auf Basis der Lageinformation eine Karte der Umgebung generiert werden kann, ist Thema des nächsten Kapitels.

Kapitel 4

Umweltmodellierung für Laufmaschinen

Combining data from several sensors and from a pre-stored model of the domain provides a way to enhance the reliability of a perception system¹.

(Crowley, 1989)

Ein Umweltmodell integriert das gesamte unter Einsatz der Sensoren gesammelte Wissen eines Roboters über die reale Welt. Sowohl die Bahnplanung als auch die vorausschauende Adaption nutzen das Umweltmodell als Wissensbasis. Somit kommt der Wahl eines geeigneten Ansatzes zur Umweltmodellierung eine zentrale Bedeutung zu. Es ist im Folgenden zu klären, auf welche Art das Wissen gespeichert werden soll und auf welche Weise die aus den Sensordaten akquirierte Information in die Umweltrepräsentation überführt werden kann. Im Rahmen von Studien- und Diplomarbeiten (Frommberger, 2002; Bär, 2005) wurden hierzu wertvolle Beiträge geleistet.

4.1 Anforderungen und Randbedingungen

Um zu einer Speicherung von Wissen zu gelangen, das auf der durch die Sensorik ermittelten Information beruht, ist es zunächst nötig, sich über die Vorgehensweise dieser Datenverarbeitung klar zu werden. Drei Hauptschritte für die Verarbeitung von Sensordaten zu verwertbaren Karten werden in Hoppen et al. (1990) zusammengestellt:

1. Merkmalsextraktion aus den Sensorrohdaten
2. Fusion der Daten der verschiedenen Sensortypen und -quellen
3. Automatische Generierung eines interpretierbaren Umweltmodells mit verschiedenen Abstraktionsebenen

Da die Sensordaten nicht alle gleichzeitig eintreffen, muss das Umweltmodell ständig mit den aktuellen Informationen aus der Sensorik abgeglichen werden. Für die Aktualisierung einer bestehenden Datenbasis nennt Crowley (1989) ebenfalls ein dreischichtiges Vorgehen:

¹Die Kombination verschiedener Sensordaten mit einem zuvor hinterlegten Modell des Einsatzgebiets bietet eine Möglichkeit, um die Zuverlässigkeit eines Wahrnehmungssystems zu steigern.

1. Aufbau einer abstrakten Beschreibung der aktuellen Sensordaten (Sensormodell)
2. Ermittlung der Übereinstimmung zwischen dem aktuellen Sensormodell und dem gegenwärtigen Inhalt des kombinierten lokalen Weltmodells
3. Veränderung der Komponenten des kombinierten lokalen Modells und Verstärkung oder Abschwächung von Zuverlässigkeitswerten, um die Übereinstimmungen wiederzugeben

4.1.1 Einschränkungen

Bei der Erstellung von Umweltmodellen unterliegt man einer Reihe von Einschränkungen (vergleiche auch Thrun (1998)). Die Problemfelder lassen sich dabei in drei Klassen einteilen:

Wahrnehmungsschwierigkeiten: Hierzu zählen alle Probleme bei der Erfassung der realen Welt wie Störerauschen, technische Sensorbegrenzungen oder nicht erfassbare Außeneinflüsse.

Abstraktionsschwierigkeiten: Eine Umwandlung der Sensordaten in ein Modell führt zu einem Wissensverlust.

Verarbeitungsschwierigkeiten: Um ein angemessenes Roboterverhalten zu ermöglichen, unterliegen alle notwendigen Verarbeitungsschritte wie Datenüberführung und Planung gewissen Echtzeitbedingungen.

Diese auftretenden Schwierigkeiten pflanzen sich in die anderen Problemebenen fort: Je schlechter die Eingangsdaten sind, desto mehr Aufwand muss in ihre Aufbereitung gelegt werden und desto schwieriger wird die Wahl eines geeigneten Modells. Je ungünstiger die Repräsentation gewählt ist, desto aufwändiger wird das Gewinnen von Strategien aus dem Datenbestand. Je mehr Aufwand jeweils betrieben werden muss, desto schwieriger ist das Einhalten der Echtzeitbedingung.

Vor allem die Folgen ungenauer oder gar falscher Sensordaten sind bis in die höchste Systemebene wahrnehmbar: Filter müssen hinter die Sensorik geschaltet werden, die Repräsentation muss mit Fehler umgehen können und die Auswertung des Wissens muss stets die Möglichkeit von Fehlinformation betrachten – all diese Maßnahmen benötigen Rechenzeit. Es empfiehlt sich also insbesondere auf eine effiziente Sensorik zu achten, um Rechenaufwand, Energie und Gewicht möglichst gering zu halten.

4.1.2 Besondere Anforderungen in unstrukturiertem Gelände

Der Schwerpunkt der folgenden Ausführungen liegt in der Erfassung von unstrukturiertem Gelände, um begehbare Pfade für Laufmaschinen zu finden. Im Vergleich zu Kartographierungsverfahren in Gebäuden und in strukturierter Umgebung sind in freiem Gelände zahlreiche einschränkende Randbedingungen zu berücksichtigen:

Unabhängigkeit von Kantenrepräsentation: Die Repräsentation natürlicher Umgebung kann nicht auf der Detektion von Punkten, Ecken und Kanten oder vergleichbarer Strukturen beruhen. Interpretierbare Karten müssen generiert werden können, auch ohne strukturelle geometrische Informationen, da nicht erwartet werden kann, dass diese überhaupt vorliegen oder abgeleitet werden können.

Weitestgehende Unabhängigkeit von a-priori-Karten: Unstrukturiertes Gelände ist oft derart inhomogen und ständigen Veränderungen unterworfen (man denke z.B. an die Vegetation), dass fein granuliert a-priori-Karten kaum möglich sind. A-priori-Karten können als zusätzliche Informationsquelle von Nutzen sein, aber das System sollte auch ohne sie navigieren können.

Robustheit gegen Lokalisationsungenauigkeiten: In natürlichen Umgebungen ist mit mehr Störeinflüssen zu rechnen als in geschlossenen Räumen, so dass die Lokalisation mit Ungenauigkeiten behaftet ist (vergleiche hierzu die Ausführungen in Abschnitt 3). Die Umweltmodellierung sollte deshalb weitestgehend robust gegen Lokalisationsungenauigkeiten sein.

Kritische Hindernisklassifikation: Vieles, was die Sensorik wahrnimmt, muss kein wirkliches Hindernis sein. Ein großer Grasbüschel beispielsweise kann von einer Laufmaschine mit Leichtigkeit überwunden werden. Wünschenswert wäre also ein kritischer Umgang mit der Klassifikation „Hindernis“.

Informationen über Wegbeschaffenheit: Für Karten in größerem Maßstab sind Informationen über die Begehbarkeit des Geländes von Nutzen, die während der Fortbewegung des Roboters gewonnen werden. Solches Wissen um „unsichere Stellen“ und „sichere Pfade“ sollte in die Kartierung der Umwelt übernommen werden.

4.1.3 Zusätzliche Anforderungen bei Laufmaschinen

Ein fundamentaler Unterschied in der Steuerung einer Laufmaschine zu radgetriebenen Robotern ist, dass nicht nur eine Lenkachse und ein Motor angesteuert werden muss, sondern eine Vielzahl von einzelnen Beinen. Genügt bei einem radgetriebenen Roboter prinzipiell der Pfad, auf dem er einen Zielpunkt erreichen muss, sind für eine Laufmaschine Steuerungsentscheidungen für jeden einzelnen Schritt notwendig. Jeder neue Fußaufsetzpunkt setzt eine separate Regelungsentscheidung voraus. Die jeweils unmittelbar bevorstehende Entscheidung einer Laufmaschine ist stets ihr nächster Schritt. Ausgehend von dieser Feststellung lassen sich folgende Anforderungen für Umweltmodelle für Laufmaschinen benennen:

Detailgenauigkeit im Nahfeld: Für die Planung der nächsten Fußaufsetzpunkte sollte die unmittelbare Umgebung detailliert in einer hohen Auflösung erfasst werden.

Gezielte Nachmessmöglichkeit: Der in Bewegungsrichtung der Maschine liegende Raum kann nicht immer vollständig und exakt kartiert werden. Eine gezielte sensorische Untersuchung im Bereich des nächsten Fußaufsetzpunktes ist sinnvoll.

Exakte Hinderniskartierung: Auf Grund der hohen Flexibilität von Laufmaschinen können Hindernisse je nach deren Größe, Form und Beschaffenheit überwunden werden. Um der Laufmaschine alle Handlungsmöglichkeiten offen zu lassen, sollten die Geometrien von Hindernissen möglichst genau erfasst werden.

Effiziente Generierbarkeit bei schmalbandiger Sensorik: Die Einschränkungen bezüglich der einsetzbaren Sensorik wurden in Abschnitt 2.1 diskutiert. Zudem ist die Lokalisation der Laufmaschine, u.a. auch durch die vielen Bewegungsfreiheitsgrade bedingt, sehr störanfällig (vergleiche Abschnitt 3). Diesen Gegebenheiten sollte das Weltmodell Rechnung tragen: Es sollte auch unter der Einschränkung schmalbandiger und nicht redundanter Sensorik leicht erstellbar sein.

Extraktion metrischer Information: Grundlegende Parameter der Steuerung vieler Laufmaschinen sind metrischer Natur wie beispielsweise die Positionierung der Füße. Deshalb sollten derartige Informationen aus dem Weltmodell extrahierbar sein.

4.2 Geometrische und topologische Karten

Es gibt zwei verschiedene Möglichkeiten zur Erstellung von Umweltkarten: den geometrischen und den topologischen Ansatz (Engelson und McDermott, 1992).

Mit Hilfe des *geometrischen Ansatzes* wird versucht, die Geometrie der Welt durch Nachbildung grundlegender geometrischer Merkmale zu beschreiben: Eine wahrgenommene Kante wird beispielsweise durch eine Kante repräsentiert. Im Fall einer universellen, videogrammetrischen Sensorik wird eine geometrische Karte eine exakte visuelle Wiedergabe der realen Welt sein – genau das ist auch der Vorteil dieses Ansatzes: Karte und Welt stehen in einer unmittelbaren, wohldefinierten Beziehung. Weiterhin sind auch Position und Orientierung des Roboters direkt in die Karte einfügbar. Ein Nachteil ist, dass es schwierig ist, eine zufrieden stellende Repräsentation zu finden, die komplexe Geometrien gut abbildet und auch mit den Unsicherheiten und Einschränkungen der Sensorik in geeigneter Weise umgehen kann. Des Weiteren ist der Nutzen der gesammelten Informationen nicht unmittelbar einsichtig, die geometrische Karte abstrahiert von der Anwendung. Zudem ist der geometrische Ansatz anfällig für Fehler, vor allem Ungenauigkeiten in der Positionsbestimmung des Roboters können fatale Effekte verursachen. Nicht zuletzt stellen geometrische Modelle hohe Anforderungen an Speicher und Rechenzeit: Die Umwelt wird in all ihrer Detailhaltigkeit aufbauend auf Sensordaten abgebildet, unabhängig vom Informationsgehalt dieser Einzelheiten.

Topologische Ansätze, die bereits 1978 von B. Kuipers eingeführt wurden, stellen die gewonnenen Beobachtungen in direkte Beziehung zu den Aktionen des Roboters. Die Welt wird z.B. durch einen Graphen dargestellt, dessen Knoten Orte der wahrgenommenen Welt sind, die mit Kanten verbunden sind, die mögliche Aktionen des Roboters darstellen (Kuipers und Byun, 1991). Dies eröffnet eine „phänomenologische Repräsentation der möglichen Interaktionen des Roboters mit der Welt“ (Engelson und McDermott, 1992). Antworten auf weitergehende Fragestellungen, wie beispielsweise

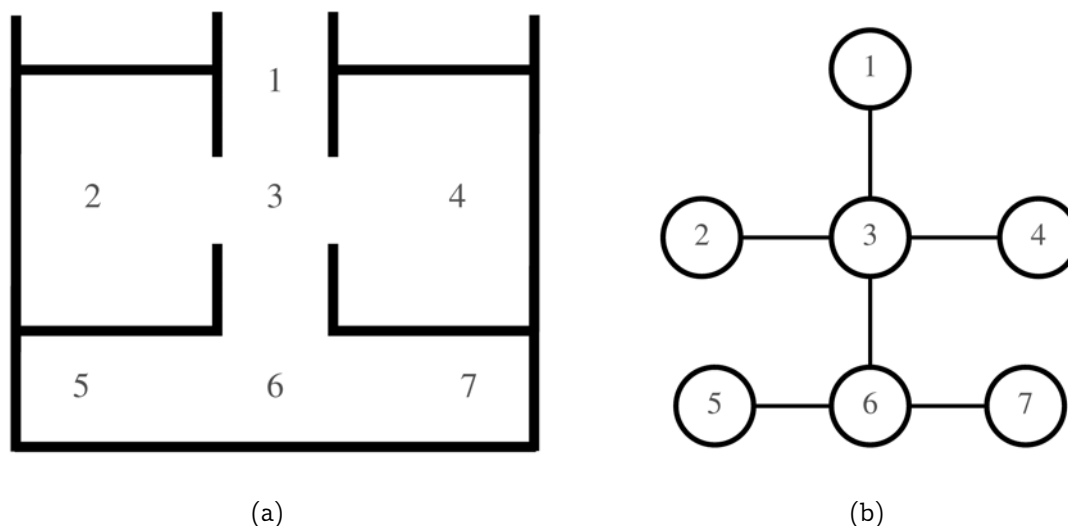


Abbildung 4.1: Beispiel einer geometrischen (a) und einer topologischen Karte (b): Die mit Nummern markierten Räume in der geometrischen Karte entsprechen den Knoten in der topologischen Darstellung, mögliche Wege sind durch Kanten repräsentiert.

Bahnplanung, lassen sich aus topologischen Karten direkt ableiten. Dadurch bleibt der Bedarf an Rechenzeit gering, zudem sind topologische Modelle auf Grund des hohen Abstraktionsgrades sparsam im Speicherverbrauch. Zur Speicherung werden vielfach Hidden-Markov-Modelle verwendet, die sich sehr effektiv und effizient auswerten lassen. Weiterhin sind topologische Modelle vergleichsweise resistent gegen Schwächen in der Odometrie des Systems, da sie statt metrischen Informationen nur Relationen zwischen Objekten der Umwelt halten. Das ist zugleich auch ihre Schwäche: Gewöhnliche Probleme wie das des Finden des „kürzesten Pfades“ lassen sich nicht ohne Weiteres lösen. Auch lässt sich die Position des Roboters nur dann aus der Karte ableiten, wenn er sich an einer „signifikanten Stelle“ befindet. Abbildung 4.1 zeigt ein Beispiel einer geometrischen und einer topologischen Karte.

Engelson und McDermott erweitern den topologischen Ansatz allerdings noch dahin gehend zur *diktiometrischen Repräsentation* (Engelson und McDermott, 1992), indem sie zu den in den Kanten gehaltenen Beziehungen zwischen den Orten geometrische Informationen hinzufügen. Die Kombination topologischer Karten und odometrischer Daten hat sich als sehr nützlich herausgestellt, auch unter der Randbedingung schwacher Odometrie (Shatkay und Kaelbling, 1997).

Es ist leicht ersichtlich, dass zur Erstellung topologischer oder gar diktiometrischer Karten ein hohes Maß an Wissen über die Umwelt erforderlich ist: ein Hindernis, das in eine solche Karte eingetragen wird, muss erst einmal aus den Sensordaten und der bisher vorhandenen Information als ein solches klassifiziert sowie Zusammengehörigkeiten von verschiedenen Messungen erkannt werden. Diese Ansätze eignen sich daher für eindeutig erfassbares, insbesondere aber auch für ein im Kontext bekanntes Terrain, aus dem a-priori-Karten erstellt werden können. Für die Aufgabe der Umweltwahrnehmung eines mobilen autonomen Roboters ohne oder mit nicht detailgenauem a-priori-Wissen sind sie weniger geeignet, da die erforderliche Menge und Genauigkeit an Wissen über die

Umgebung, vor allem unter Einsatz schmalbandiger, nicht redundanter Sensorik, nicht erworben werden kann.

Für geometriebasierte Umweltmodellierungsansätze gibt es allerdings, außer den bereits genannten, auch solche Einschränkungen, die Alberto Elfes als die „Lücke zwischen zwei Informationsebenen“ bezeichnet (Elfes, 1989). Einerseits existiert die Ebene der unpräzisen und limitierten Sensordaten, andererseits die Ebene des abstrakten symbolischen und geometrischen Weltmodells, auf dem operiert wird und aus dem Information für das weitere Verhalten des autonomen Systems gewonnen wird. Elfes folgert daraus, dass geometrische Ansätze vor allem in gut strukturierten Umgebungen sinnvoll sind, für komplexere Szenarien aber nur beschränkte Eignung aufweisen. Topologische Karten haben damit allerdings auch Probleme, nur auf einer anderen Ebene: Mit der Komplexität der Umgebung steigt auch die der Karte und damit auch ihr Erstellungs- und Auswertungsaufwand.

Die wichtigsten Eigenschaften beider Ansätze bzgl. verschiedener Kriterien sind in Tabelle 4.1 zusammengefasst. Offensichtlich stehen die Vor- und Nachteile der Methoden jeweils orthogonal zueinander. Es empfiehlt sich also, beide Ansätze zu kombinieren, um die jeweiligen Stärken ausnutzen zu können. Eine nahe liegende Möglichkeit ist, auf der Grundlage einer geometrischen Karte eine topologische Karte zu erzeugen, indem man erstere nach kohärenten Regionen durchsucht und daraus eine Topologie generiert (Thrun und Bücken, 1996). Simhons und Dudek (1998) erstellen topologische Karten, indem sie lokale geometrische Karten („islands of reliability“) in Relation zueinander setzen. Solche Vorgehensweisen lassen sich unter „*first metric, then topological*“ zusammenfassen. Auch eine umgekehrte Vorgehensweise („*first topological, then metric*“) wird praktiziert: Aus topologischen Karten kann mit Hilfe des EM-Algorithmus die wahrscheinlichste geometrische Repräsentation abgeleitet werden (Thrun et al., 1998) (vergleiche hierzu auch Abschnitt 3.2.4). Beide Ansätze erreichen das gleiche Ziel: Durch die gleichzeitige Verwendung zweier Karten kann sichergestellt werden, dass für jede auftauchende Problemstellung eine geeignete Umweltrepräsentation verfügbar ist.

Ein weiterer Ansatz zur Kombination topologischer und geometrischer Informationen, die *Variable Resolution Maps* (Arleo et al., 1999), wird in Abschnitt 4.6.2 behandelt.

4.3 Belegtheitsgitter

Folgt man Alberto Elfes in seiner Argumentation (Elfes, 1989), so gibt es keinen geeigneten Ansatz, unstrukturiertes Gelände zu repräsentieren. Schon die sprachlichen Bezeichnungen spiegeln die Schwierigkeit wider: Wie bekomme ich Unstrukturiertes in eine *Datenstruktur*? Wenn sich ein geeigneter Ansatz nicht finden lässt, stellt sich die Frage nach dem am wenigsten ungeeigneten Ansatz.

Elfes selbst führt als Antwort auf die von ihm benannten Einschränkungen geometrischer Weltmodelle das *Belegtheitsgitter* (*Occupancy Grid*, seltener auch *Certainty Grid*) ein, welches ebenfalls von Moravec propagiert wurde (Moravec, 1988). Es dient der Repräsentation der Umwelt vor allem für unsichere und lückenhafte Sensorik. Die

	Geometrische Karten	Topologische Karten
Generierung	leicht zu erstellen und aktualisieren, Sensordaten sofort nach Messung eintragbar	viele Sensordaten und hohes Maß an Datenverarbeitung notwendig, Schwierigkeiten bei sehr großen Umgebungen
Speicherbedarf	extrem hoch	eher gering, abhängig von Komplexität der Umgebung
a-priori-Wissen	nützlich, aber nicht notwendig	nur bei sehr leistungsstarker Sensorik verzichtbar
komplexe Welt	Kartengröße unabhängig von Komplexität, aber Erkenntnisgewinn aus Karte aufwändig	erzeugen sehr komplexe Karten
Roboterposition	hohe Genauigkeit bei der Erfassung und ständige Korrektur notwendig	nicht nötig \Rightarrow robust gegen Störungen wie Rutschen etc.
Wiedererkennung	absolute Position des Roboters kann direkt in Karte eingeordnet werden, Orte so leicht wiedererkannt werden	wiedererkennen von Orten schwierig, vor allem, wenn sie auf verschiedenen Pfaden erreicht wurden; Probleme mit mobilen Objekten
Pfadplanung	sehr hoher Verarbeitungsaufwand, kürzester Pfad aber leicht ermittelbar	Pfadplanung direkt aus Karte ableitbar, aber evtl. suboptimale Wege durch fehlende metrische Information

Tabelle 4.1: Gegenüberstellung der Eigenschaften geometrischer und topologischer Karten: Vor- und Nachteile stehen orthogonal zueinander.

reale Welt wird dabei in ein zwei- oder dreidimensionales Gitter eingeteilt, in dem jede Gitterzelle alle in ihr enthaltenen Punkte im Raum repräsentiert. Die kontinuierliche Umgebung wird also diskretisiert: Jedem wirklichen Punkt wird eine Gitterzelle zugeordnet, die eine probabilistische Aussage über den Zustand der ihr zugehörigen realen Region speichert. Dieser Zustand der Zelle entspricht in der vorgestellten Form ihrer Belegtheitsinformation, die entweder die binäre Information „frei oder belegt“ sein kann oder, wie üblicherweise verwendet, ein Wert $P(c = \text{occ}) \in [0, 1]$, der die Wahrscheinlichkeit angibt, mit der die Zelle belegt (*occupied*) ist. Gitter, die noch weitergehende Daten als den Belegtheitsstatus halten, nennt Elfes *Inferenzgitter*.

Bei der Aktualisierung der Zellinhalte wird aus einer Anzahl von Sensormessungen eine jeweils neue Belegtheitswahrscheinlichkeit nach dem Satz von Bayes errechnet. Ausgehend von einer Reihe von Sensormessungen $\{r\}_t = \{r_0, r_1, \dots, r_t\}$ und der daraus geschätzten Belegtheitswahrscheinlichkeit für die Zelle c , $P(s(c) = \text{occ} | \{r\}_t)$ kann eine neue Schätzung $P(s(c) = \text{occ} | \{r\}_{t+1})$ nach einer weiteren Messung r_{t+1} rekursiv gewonnen werden:

$$P(s(c) = \text{occ} | \{r\}_{t+1}) = \frac{p(r_{t+1} | s(c) = \text{occ}) P(s(c) = \text{occ} | \{r\}_t)}{\sum_{s(c)} p(r_{t+1} | s(c)) P(s(c) | \{r\}_t)}$$

Die Wahrscheinlichkeitsverteilung $p(r_{t+1} | s(c))$ ergibt sich aus dem Sensormodell, die Prior-Wahrscheinlichkeit $P(s(c) | \{r\}_t)$ kann direkt aus dem Belegtheitsgitter abgelesen werden. Betrachtet werden dabei alle Zellen c im Erfassungsbereich der jeweiligen Messung. Dieser wahrscheinlichkeitstheoretische Ansatz erklärt sich auch aus den zu der Zeit der Einführung der Belegtheitsgitter vorwiegend verwendeten Ultraschallsensoren. Diese weisen einen hohen Grad an Unsicherheit auf, zu gewinnen sind nur probabilistische Informationen über eventuelle Hindernisse. Dafür ist der Bayessche Ansatz durchaus gut geeignet: Die aus dem Sensormodell hervorgehende Wahrscheinlichkeitsinformation der Sensormessung kann direkt in die Gitterrepräsentation eingehen.

Diese Methode ist relativ rechenzeitaufwändig und setzt voraus, dass der Roboter zum Aufnehmen der Messungen stehen bleibt. Borenstein und Koren (1991) stellten daher als Alternative dazu das *Histogramm-Gitter* (*Histogram Grid*) vor. Dabei wird für jede Sensormessung nur genau eine (die „anscheinend getroffene“ Zelle) aktualisiert, indem ein Belegtheitswert $b \in \mathbb{N}$ hochgezählt wird. Dabei ergibt sich eine „Pseudo-Wahrscheinlichkeitsverteilung“ durch schnell aufeinander folgende Messungen, während der Roboter sich kontinuierlich weiter bewegt. Diese Parallelisierung von Roboterbewegung, Umweltmodellierung und Planung, die die starre Abfolge „Stehenbleiben – Wahrnehmen – Planen“ ersetzt, ist neben der einfachen Arithmetik der Vorteil dieser Methode. Es lässt sich zeigen, dass diese Herangehensweise exaktere Karten generiert als der wahrscheinlichkeitstheoretische Ansatz (Rascke und Borenstein, 1990).

Die Histogramm-Gitter haben in den 90er Jahren gegenüber den stochastischen Belegtheitsgittern zunehmend an Bedeutung gewonnen, vor allem für Sensorik mit einem relativ niedrigen Grad an Unsicherheit wie Laser-Scanner und Infrarotsensoren.

Ein entscheidender Vorteil der Idee der Belegtheitsgitter und dessen Varianten ist, dass sich die Daten verschiedener Sensoren leicht in einer Repräsentation vereinen las-

sen. Ist die Position eines Punktes im Raum (bzw. eine diesbezügliche Wahrscheinlichkeitsverteilung) bekannt, können mehrere solche Informationseinheiten mittels einfacher Arithmetik zusammengefasst werden.

4.4 Erste Schlussfolgerungen

Nachdem in den letzten Abschnitten sowohl mehrere Anforderungen an das zu entwerfende Navigationssystem gestellt als auch die Grundzüge sich anbietender Konzepte erörtert wurden, kann nun eine erste Entwurfs-Entscheidung getroffen werden. Offensichtlich ist der topologische Kartierungsansatz trotz einiger deutlicher Vorteile für Laufmaschinen in natürlicher Umgebung nicht geeignet. Daher werden im Folgenden geometrische Kartierungsmethoden vertieft. Im Einzelnen sprechen folgende Gründe für diese Entscheidung:

1. Die Generierung eines zufrieden stellenden topologischen Modells ist mit der verfügbaren schmalbandigen nichtredundanten Sensorik nicht möglich. Eine geeignete Klassifizierung beobachteter Strukturen setzt eine ausreichende Dichte an Sensordaten voraus. Der Roboter sollte sich aber auch auf seine Weltmodellinformationen verlassen können, auch wenn nur vereinzelte Sensormessungen verfügbar sind. Zum Beispiel kann ein Mittel- oder Hinterbein der Maschine aus den von der Sensorik des Vorderbeines gewonnenen Daten ausreichend Information für seinen nächsten Schritt ziehen, auch wenn diese nur einen sehr geringen Bereich der Welt umfasst. Eine verlässliche topologische Repräsentation ist in diesem Fall weder möglich noch sinnvoll.
2. Liegen dem System a-priori-Informationen über die Umwelt vor, können diese Daten durch Abgleich mit den eingehenden Sensordaten verwehrt werden und somit Modellwissen bei Sensorunsicherheiten nutzen. Für Laufmaschinen in natürlicher Umgebung liegt aber in der Regel kein Modellwissen der Umwelt vor.
3. Die Ansteuerung vieler Laufroboter erfolgt über messbare Parameter wie Schrittlänge oder Schritthöhe. Zu ihrer Ermittlung ist es notwendig, kontinuierlich metrische Informationen aus dem Weltmodell zu gewinnen. Reine topologische Ansätze beinhalten diese Informationen nicht, bestenfalls kämen diktiometrische Karten in Betracht.
4. Eine detaillierte Kartierung des Untergrunds ist mit topologischer Repräsentation nur dann möglich, wenn dieser überwiegend plan ist. Ansonsten müsste jede Bodenunebenheit durch einen eigenen Knoten repräsentiert und in Bezug zu anderen Unebenheiten gesetzt werden, was im Endeffekt zu einer quasi-geometrischen Darstellung (nur mit ungleich höherem Aufwand) führen würde. Gerade das Finden von geeigneten Fußaufsetzpunkten ist eine Kernaufgabe für Laufmaschinen, so dass auf eine hoch aufgelöste lokale Bodenrepräsentation nicht verzichtet werden kann.

Tabelle 4.2 stellt die Vor- und Nachteile der vorgestellten Ansätze gegenüber.

	geometrisch	Belegtheitsgitter	topologisch	diktiometrisch
Allgemein:				
einfache Sensorik	○	++	–	–
Rechnereffizienz, Echtzeitfähigkeit	○	○	–	--
Unstrukturiertes Gelände:				
unabhängig von Kantenrepräsentation	–	○	–	–
unabhängig von a-priori-Karten	○	○	–	–
robust gegen Lokalisationsungenauigkeiten	–	–	+	○
kritische Hindernisklassifikation	○	+	–	–
Laufmaschinen:				
detailgenau im Nahfeld	+	+	--	--
Möglichkeit zum Nachmessen	+	++	–	○
exakte Hinderniskartierung	+	+	--	--
metrische Information extrahierbar	++	+	--	+

++: gut geeignet +: geeignet ○: mäßig geeignet -: wenig geeignet --: ungeeignet

Tabelle 4.2: Zusammenstellung der Eignung der vorgestellten Kartierungsmodelle unter Berücksichtigung der formulierten Anforderungen: Topologische Karten erweisen sich für den Einsatz auf Laufmaschinen als ungeeignet. Auch geometrische Karten zeigen in unstrukturiertem Gelände Schwächen auf. Einige lassen sich durch die Repräsentation im Belegtheitsgitter abmildern.

4.5 Dimensionsaspekte

Im Folgenden wird diskutiert, ob eine 2D- oder 3D-Repräsentation der Umwelt für Laufmaschinen erforderlich ist. Letzteres erfordert einen höheren Speicherbedarf und damit ein Anstieg des Verarbeitungsaufwands.

In der Robotik werden überwiegend zweidimensionale Kartierungsansätze verfolgt. Oft reicht dies auch aus: Für einen mobilen Putzroboter genügt ein Grundriss seines Einsatzgebiets, um die dort eingezeichneten Hindernisse umgehen zu können. Informationen über die dritte Dimension sind dabei ohne Belang: Es ist für seine Aufgabe nicht relevant, wie hoch ein im Weg stehender Schrank ist oder ob der Fußboden uneben ist. Entscheidend ist, ob er die zu reinigenden Flächenelemente erreichen kann oder nicht. Die Information „Hindernis oder nicht“ ist ausreichend.

Für Laufmaschinen ist dies nicht der Fall. Hindernisse sollen nicht nur vermieden, sondern möglichst überwunden werden. In Abschnitt 4.1 wurden zwei Anforderungen genannt, die ein zweidimensionales Umweltmodell für die Arbeit mit Laufmaschinen ausschließen:

1. Die Forderung nach exakter Hinderniskartierung zieht die Notwendigkeit einer möglichst exakten Wiedergabe der Geometrie des Hindernisses nach sich. Höheninformationen müssen gespeichert werden, damit die Maschine entscheiden kann, ob sie eine Barriere übersteigen kann oder nicht.
2. Detailgenauigkeit im Nahfeld bedeutet insbesondere, dass der Untergrund derart erfasst ist, dass der nächste Fußaufsetzpunkt möglichst genau ermittelt werden kann. Auch dazu ist die Speicherung von Höheninformation notwendig.

Eine weit verbreitete Methode ist es, in einem zweidimensionalen Gitter für jede Zelle eine Höheninformation und ggf. weitere Daten wie Belegheitswahrscheinlichkeit zu speichern (vergleiche beispielsweise Murphy et al. (2002)). Solche Ansätze werden oft mit der Bezeichnung $2\frac{1}{2}$ D-Modell klassifiziert. Ihre Verwendung setzt voraus, dass die Höheninformation jeweils bestimmbar ist, was eine geeignete Sensorik voraussetzt. Mit $2\frac{1}{2}$ D-Modellen lassen sich allerdings keine konkaven Hohlobjekte oder Hindernisse wie Tore, Balken, Tische beschreiben, die von einer Laufmaschine entweder unter- oder überquert werden kann. Insofern ist die Verwendung eines 3D-Modells in Form eines Belegheitsgitters ein durchaus geeigneter Ansatz. Er wird im Folgenden eingehend diskutiert.

4.6 Die Repräsentation der Umwelt

Im letzten Abschnitt wurde auf die Notwendigkeit einer dreidimensionalen Repräsentation der Umwelt und somit auf die Verwendung eines Volumenmodells hingewiesen. Solche Ansätze weisen eine hohe Komplexität auf und erfordern erheblichen Speicherplatz. Eine Speicherung größerer Umgebungen ist allerdings nicht notwendig: Für Missionen, die weit reichende Karten benötigen, sind für Laufmaschinen zweidimensionale Umweltrepräsentationen mit geeigneten Wegbeschaffens- und Höheninformationen ausreichend

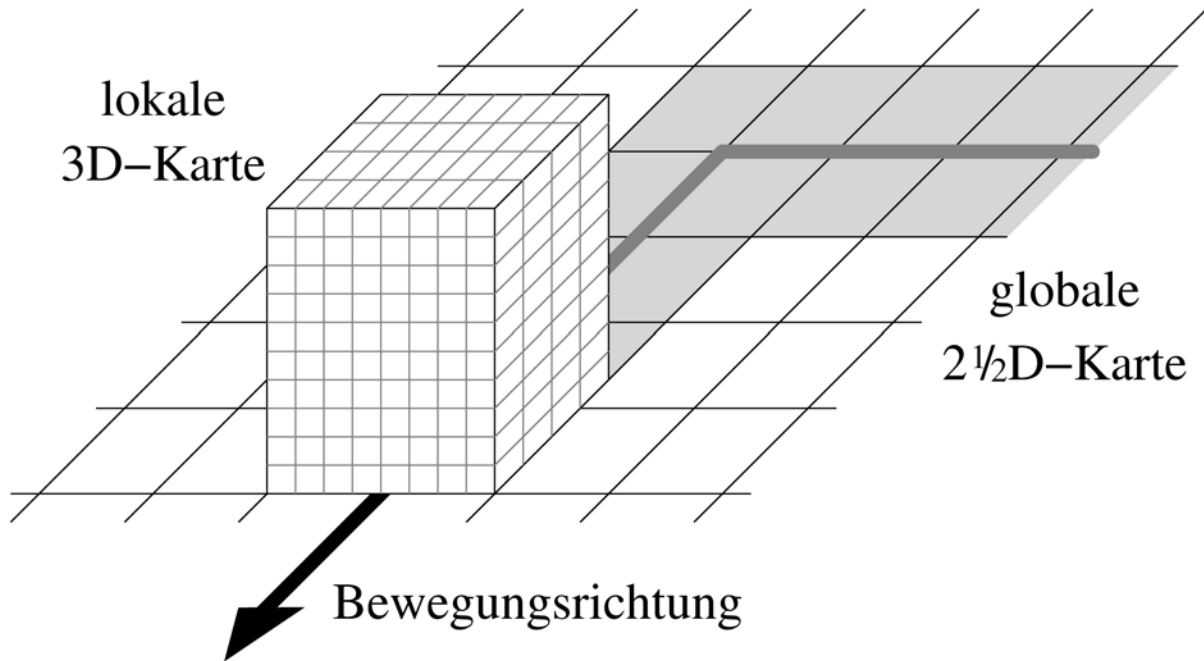


Abbildung 4.2: Schematische Darstellung: Verschieben der 3D-Umgebung und Erzeugung einer $2\frac{1}{2}$ D-Karte. Die grauen Bereiche wurden aus der sich mit dem Roboter „bewegenden“ Urkarte generiert.

(vergleiche hierzu die Kapitel 5 und 6). 3D-Daten benötigt man für das Nahfeld; sie sind erforderlich, um Höheninformationen aus verrauschten Sensordaten zu extrahieren. Es bietet sich demnach an, die unmittelbare Umgebung eines Roboters in 3D und alles weitere in $2\frac{1}{2}$ D zu speichern. Damit wird auch der in Abschnitt 4.1 genannten Forderung von Hoppen et al. (1990) entsprochen, ein Modell mit verschiedenen Abstraktionsebenen zu generieren. Die einzelnen Karten besitzen den Detaillierungsgrad, der für die jeweilige Verarbeitungsaufgabe notwendig ist. Die dreidimensionale Repräsentation soll im Folgenden als *Urmodell* oder *Urkarte* bezeichnet werden.

Aufbauend auf einem Ansatz von Murphy et al. (2002) wird die unmittelbare Umgebung des Roboters 3-dimensional repräsentiert. Dieses Volumen wird mit der Bewegung des Roboters verschoben. Die weitere Darstellung der Roboterumgebung bleibt in einer 2D-Repräsentation, welche aus der 3D-Repräsentation generiert wird (Abbildung 4.2). Der Speicherbedarf solcher Übersichts-Karten ist unkritisch.

Das Verschieben der unmittelbaren Umgebungskarte bedeutet allerdings, dass das erworbene Wissen in den durchlaufenen Bereichen wieder verloren geht. Dies ist aber vertretbar: Der Wert detailgenauer Informationen nimmt mit der Zeit durch die auftretenden Ungenauigkeiten in der Lagebestimmung der Maschine stetig ab, so dass ausgiebige Neuvermessungen ohnehin notwendig sein werden, wenn der Roboter nach längerer Zeit an eine bereits bekannte Stelle zurückkehrt. Der Speicheraufwand des Vorhaltens steht in keinem Verhältnis zu dem zu erwartenden Nutzen. Wenn man die detaillierten Umweltinformationen speichern möchte, wäre es eine Option, sie auf einen Datenträger wie z.B. einer Festplatte auszulagern.

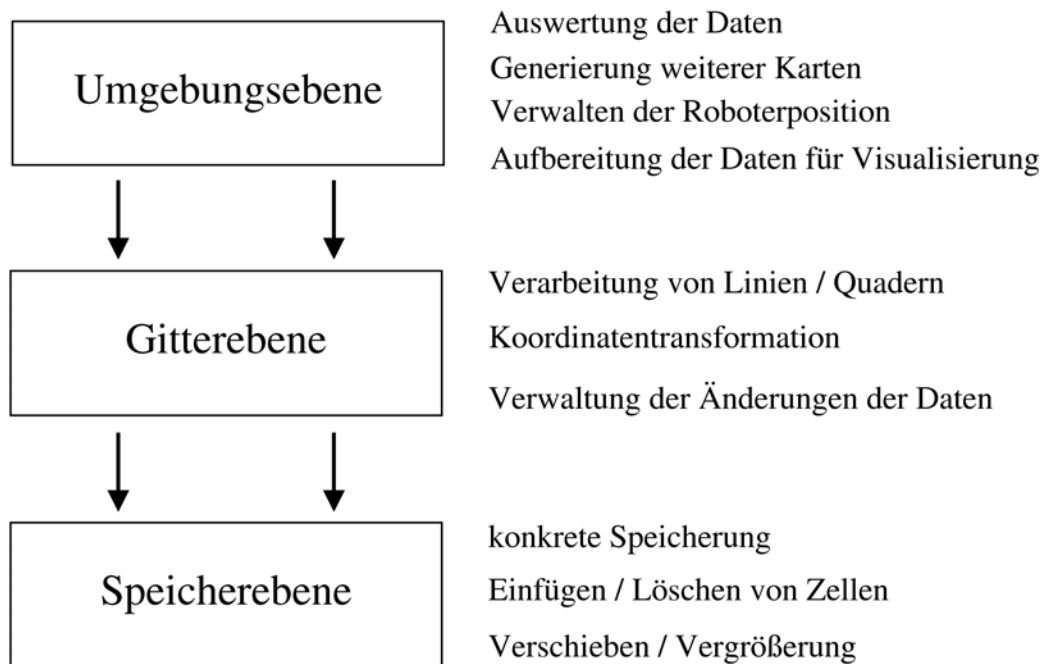


Abbildung 4.3: Schichtweiser Aufbau der Karten des Umweltmodells mit den entsprechenden Aufgaben.

4.6.1 Datenkapselung mit Hilfe eines Schichtenmodell

Für die Repräsentation der Umwelt wird im Folgenden ein Schichtenmodell vorgeschlagen, das für alle in diesem Weltmodell verwendeten Karten angewendet wird. Der Schichtenansatz folgt dem Paradigma der objekt-orientierten Kapselung, wonach alle Details eines Objektes versteckt werden sollen, die nur seine inneren Vorgänge betreffen.

Danach besteht jede der verwendeten Karten aus drei Ebenen, die durch fest definierte Schnittstellen miteinander verbunden sind. Den Aufbau und die jeweiligen Aufgaben zeigt Abbildung 4.3. Im einzelnen sind dies, beginnend mit der untersten Schicht:

Speicherebene: In dieser Schicht erfolgt die konkrete Speicherung der Umweltdaten in einer Datenstruktur. Hier sind die Operationen definiert, die direkt den Datenbestand des Umweltmodells betreffen: Einfügen, Löschen und Auslesen der Zellen, Vergrößern des Definitionsraums und Verschieben der Karte.

Gitterebene: Diese Schicht liefert eine Sicht auf das Belegtheitsgitter. Reale Koordinaten werden in das interne Koordinatensystem der Speicherebene transformiert und Methoden zur Verarbeitung größerer geometrischer Primitive wie Linien und Quadern werden zur Verfügung gestellt. Zudem erfolgt hier eine Verwaltung der letzten Änderungen am Datenbestand; dies wird beispielsweise für Visualisierungszwecke verwendet.

Umgebungsebene: Dies ist die höchste Schicht im Umweltmodell. Neben der Verwaltung der Position des Roboters in der Karte werden hier die Funktionen zur

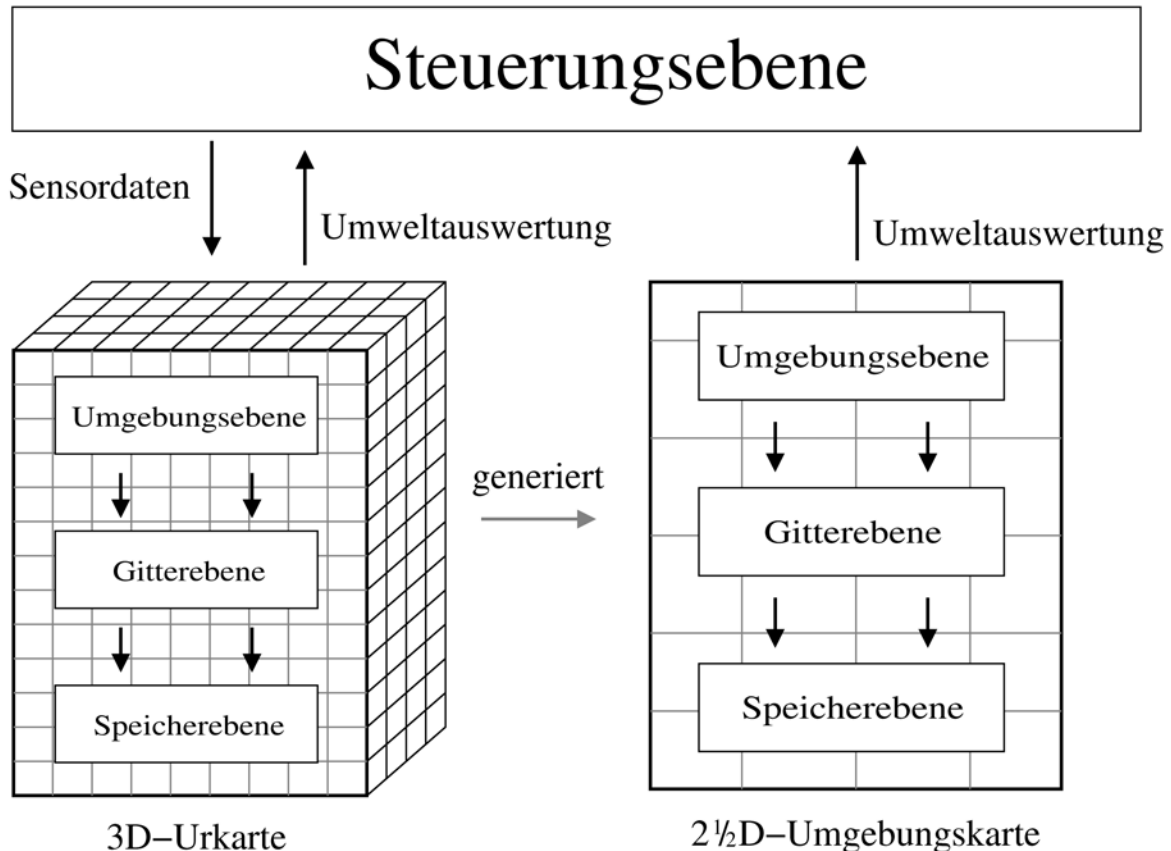


Abbildung 4.4: Interaktion von Steuerungsebene und Karten verschiedener Abstraktionsstufen: Die Steuerungsebene sendet Daten nur zur Urkarte, empfängt aber Auswertungen von allen Karten. Die Erzeugung von Unterkarten erfolgt in den Karten selbst.

Auswertung des Weltmodells bereitgestellt, beispielsweise Generierung von Karten anderer Abstraktionsebenen, Finden von Fußaufsetzpunkten usw. Zudem ist die Umgebungsebene für die Aufbereitung der Visualisierungsdaten und deren Transport zur grafischen Oberfläche verantwortlich.

Die Vorteile dieser Architektur im praktischen Einsatz sind wie folgt: Durch die Kapselung und die Definition definierter Schnittstellen kann jede Schicht ausgetauscht werden. Es ist also beispielsweise problemlos möglich, die Realisierung der Speicherebene durch eine effizientere Variante zu ersetzen, ohne am Rest der Implementierung Änderungen vornehmen zu müssen. Zudem sichert dieser Ansatz hohe Wiederverwertbarkeit des Programmcodes: Die Programm-Klassen für Gitter- und Speicherebene werden in der vorliegenden Umsetzung des Weltmodells nicht nur für alle Versionen von Karten, sondern auch gleichermaßen für die Realisierung der Visualisierungssoftware verwendet.

Allen Karten übergeordnet ist eine *Steuerungsebene*. Sie gibt die aufbereiteten Sensordaten an die 3D-Repräsentation (und nur an diese) weiter, empfängt aber Auswertungen der Umweltdaten von allen generierten Karten und verarbeitet diese entsprechend (Abbildung 4.4).

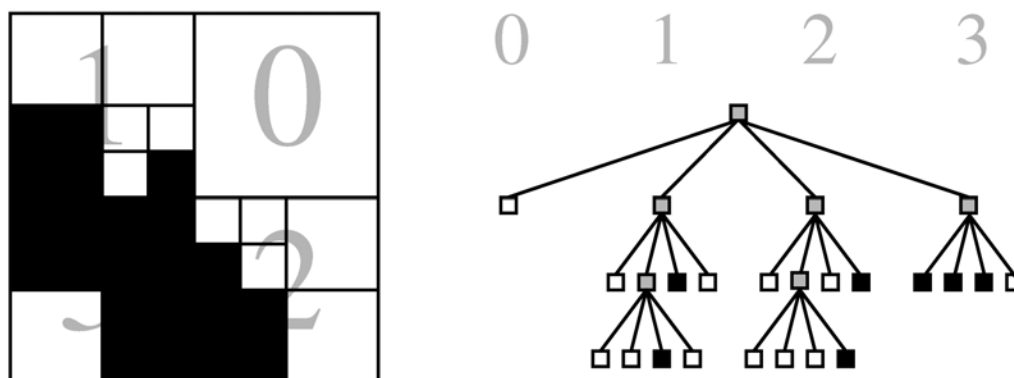


Abbildung 4.5: Repräsentation einer Unterteilung einer Fläche in einem Quadtree. Belegte Knoten sind schwarz, freie weiß und gemischte grau dargestellt.

4.6.2 Speicherung in einer Octree- und Quadtree-Datenstruktur

Eines der Hauptprobleme bei geometrischen Modellierungsansätzen wie dem 3D-Belagtheitsgitter ist der hohe Speicherbedarf. Für die Speicherung in einem Array wären bei einer Zellgröße von 5 cm und einer Ausdehnung der Karte von 10 m beispielsweise 8 Millionen Gitterzellen zu speichern. Dieser Umstand wird noch dadurch erschwert, dass die Implementierung des Weltmodells auf einem auf dem Roboter befindlichen Rechner laufen soll, dessen Ressourcen begrenzt sind. Eine Speicherung der Weltmodell-Daten in einem herkömmlichen Array ist daher kritisch.

Eine bewährte Methode zur platzsparenden Speicherung von Volumendaten ist die Nutzung von *Octrees* (Achterbäumen). Diese Datenstruktur stellt eine rekursive Unterteilung eines (kubischen) Raumes in jeweils acht Oktanten dar, die durch einen Baum mit acht Söhnen für jeden Knoten repräsentiert wird (Jackins und Tanimoto, 1980). Die Oktanten werden dabei sukzessive solange unterteilt, bis ihr Inhalt homogen ist. Somit eignen sich Octrees insbesondere zur Speicherung spärlich verteilter 3D-Daten, da in diesem Fall homogene leere Regionen vorherrschen, wenig Unterteilungen nötig sind und wenig Speicher belegt werden muss. Dies ist für die Speicherung von 3D-Belagtheitsgittern gut geeignet, da die empfangenen Sensordaten im Normalfall räumlich begrenzt sind und weite Teile der realen Welt nicht betrachtet werden (beispielsweise alles unterhalb des wahrgenommenen Bodens). Das Analogon eines Octrees für den zweidimensionalen Fall ist ein *Quadtree* (Viererbaum) mit jeweils vier Söhnen. Quadrees werden für die Speicherung der zu generierenden zweidimensionalen Karten verwendet.

Die Knoten eines herkömmlichen Octrees kennen drei Zustände: belegt, frei und gemischt. Die ersten beiden Arten von Knoten repräsentieren homogene Regionen und sind somit stets Blätter, gemischte Knoten haben jeweils wieder acht Söhne. Abbildung 4.5 veranschaulicht dieses Konzept an einem Quadtree.

In der Umsetzung steht an der Stelle eines belegten Knotens nicht nur die Information „belegt“, sondern eine Reihe von Daten, die die so repräsentierte Gitterzelle betreffen (siehe Abschnitt 4.7.2). Zwei Gitterzellen werden im Allgemeinen nie den gleichen In-

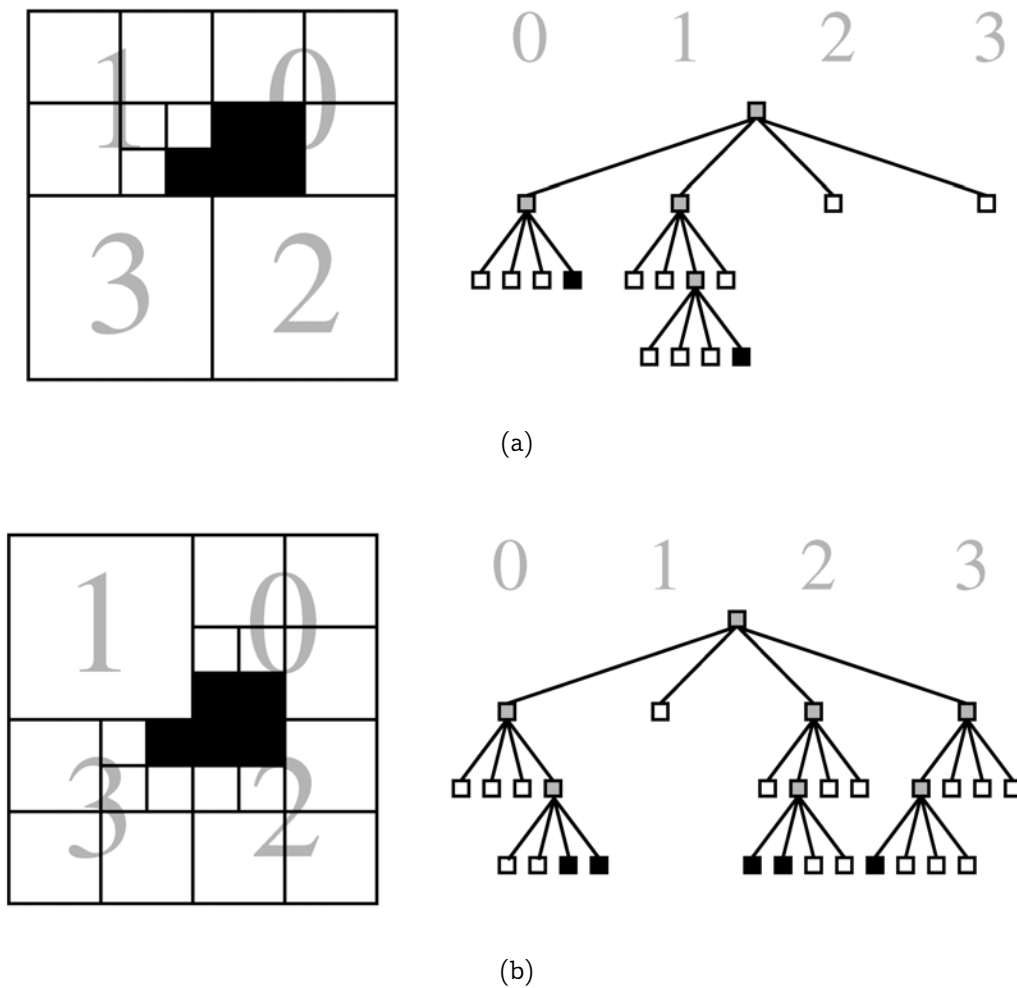


Abbildung 4.6: Speicherbedarf einer Szene in einem Quadtree: Abhängig von der Wahl des Ursprungs benötigt eine identische Szene unterschiedlich viele Knoten (a) Speicherbedarf: 17 Knoten (b) Speicherbedarf: 29 Knoten.

halt haben, so dass es keine homogenen belegten Gebiete geben kann. Die Information, sofern es welche gibt, betrifft stets eine Einheit des kleinsten Volumenelements und befindet sich somit stets in Blättern auf der Ebene der maximalen Tiefe des Octrees.

Die Unterteilung des Raumes findet bei einem Octree stets in der Mitte der Koordinatenachse jedes Oktanten statt, unabhängig von der Ausprägung der zu repräsentierenden Umgebung. So hängt die Anzahl der nötigen Unterteilungen und somit der Speicherbedarf von der willkürlichen Wahl des Ursprungs des Octrees ab (Abbildung 4.6). Eine flexiblere Unterteilung der Umwelt liefert der *Parti-Game-Algorithmus* (Moore und Atkeson, 1995). Nachteilig daran ist, dass die Gesamtgröße der Umgebung im Vorfeld bekannt sein muss und das Verfahren auf Grund seines hohen Rechenaufwands der Echtzeit-Bedingung nicht genügt. Diese Einschränkungen bringt ein anderer Ansatz, die Verwendung von *Variable Resolution Maps* (Arleo et al., 1999), nicht mit sich. Diese unterteilen die räumliche Repräsentation anhand detektierter Kanten, um eine optimale Ausnutzung von Homogenitäten der Umgebung zu erhalten. Dies stellt

eine Variante des in Abschnitt 4.2 erwähnten Ansatzes der Generierung topologischer Karten aus geometrischen Informationen („first metric, then topological“) (Thrun und Bücken, 1996) dar, da die erzeugte Repräsentation in einem Beziehungsgraph zwischen homogenen Regionen resultiert. Allerdings geschieht dies hier in Echtzeit während der Fortbewegung des Roboters, und die Ursprungsinformation wird gleich nach Verarbeitung wieder verworfen. Das größte Problem dieses Ansatzes ist seine Beschränkung auf isoorientierte Kanten, so dass das Verfahren bislang nur für die Navigation in Gebäuden verwendet wird, beispielsweise in Cho et al. (2001).

Zur Repräsentation eines Weltmodells für Laufmaschinen sind diese Ansätze nicht geeignet. Die genannten Einschränkungen von Octrees bzgl. der Speicheruneffizienz müssen deshalb hingenommen werden.

4.7 Die dreidimensionale Urkarte

Dieser Abschnitt befasst sich nun mit dem im vorangegangenen Abschnitt beschriebenen Teil-Modell, das die unmittelbare Umgebung der Maschine dreidimensional abbildet. Zunächst wird diskutiert, welche Information sinnvollerweise gehalten werden soll, um anschließend eine Methode zum geeigneten Einfügen von Sensor-Informationen in das bestehende Modell vorzustellen. Als Repräsentation wurde eine Variante eines Belegtheitsgitters gewählt (für eine Übersicht siehe Abschnitt 4.3).

4.7.1 Das erweiterte Inferenz-Gitter

Im Folgenden wird eine Variante eines Belegtheitsgitters vorgeschlagen, dass als *Erweitertes Inferenz-Gitter* bezeichnet werden soll. *Inferenz-Gitter* hat Alberto Elfes solche Belegtheitsgitter genannt, die mehr Information enthalten als nur die Belegtheitswahrscheinlichkeit (Elfes, 1989). Das vorgeschlagene Gitter verwendet als Inhalt der Zellen eine genau auf den jeweiligen Roboter und seine Sensorik zugeschnittene Datenstruktur, die neben der Wahrscheinlichkeit eines Hindernisses auch Information über die Entstehung der Messungen speichert.

Die Erweiterung besteht nun im Vorgehen bei der Zellinhalt-Modifikation. Das erweiterte Inferenz-Gitter verwendet prinzipiell den Ansatz des Histogramm-Gitters: Für jede Sensormessung wird nur eine einzige Zelle modifiziert (Borenstein und Koren, 1991). Hier wird jedoch nicht einfach ein Wert um einen festen Betrag erhöht oder vermindert, sondern die neue Messung geht mit einem Faktor ein, der sich sowohl nach einer Schätzung der Güte der aktuellen Messung als auch nach der Historie der Zusammensetzung des bestehenden Zellinhalts richtet. Damit soll insbesondere der Tatsache Rechnung getragen werden, dass die eine Zelle betreffenden Informationen zu sehr unterschiedlichen Zeitpunkten von unterschiedlichen Sensoren und unter unterschiedlichen Rahmenbedingungen akquiriert wurden.

4.7.2 Die Gitterzelle

Die Gitterzelle ist das zentrale Speicherelement des 3D-Weltmodells. In jeder Gitterzelle wird sowohl gespeichert, was an Wissen über diesen Bereich der Umwelt relevant erscheint, als auch „Metadaten“, die zur Bewertung und Einordnung der Informationen herangezogen werden können. Als solche Metadaten kann beispielsweise Wissen darüber gehalten werden, auf welche Weise die in der Zelle gespeicherten Informationen akquiriert wurden.

Die wichtigste Information in einem Belegtheitsgitter ist, was über die durch die Zelle repräsentierte wirkliche Welt ausgesagt werden kann. An solchen unmittelbaren Informationen wird in der 3D-Gitterzelle Folgendes gespeichert:

Belegtheitswahrscheinlichkeit: $\text{occ}(c) \in [0, 1]$ bezeichnet die Sicherheit, mit der die Zelle c belegt ist.

Zuverlässigkeit: $\text{cred}(c) \in [0, 1]$ gibt eine Wahrscheinlichkeit an, mit der die für diese Zelle gespeicherten Daten glaubwürdig sind.

Es stellt sich die Frage, wieso mit $\text{cred}(c)$ neben $\text{occ}(c)$ noch ein weiterer stochastischer Wert gehalten wird und wo der Unterschied liegt. Als Hinleitung zu dieser Entscheidung mag das folgende Beispiel dienen:

Beispiel 4.1 *Eine Tischkante rage genau zur Hälfte in die zu betrachtende Gitterzelle hinein; dazu nehmen wir an, man würde nur die Belegtheitswahrscheinlichkeit speichern. Ideale Sensorwerte vorausgesetzt würde man $\text{occ}(c) = 0,5$ erwarten, da die Hälfte der Messungen ein Hindernis erfassen würde, die andere Hälfte nicht. Die 0,5 würden also genau das Richtige aussagen: dass die Zelle zur Hälfte belegt ist. Fällt aber die Voraussetzung idealer Sensorik weg, lässt sich diese Aussage nicht mehr halten, da genauso gut eine komplett belegte Zelle und 50% Fehlmessungen vorliegen könnten. Der Schwachpunkt der Messungen ist also nicht ersichtlich – es kann sowohl an schwacher Sensorleistung als auch an der nicht gittertauglichen realen Welt liegen.*

Die Aufspaltung in eine Belegtheitswahrscheinlichkeit und Zuverlässigkeit behebt diesen Schwachpunkt. $\text{occ}(c)$ liefert eine Beschreibung der Umwelt am Ort c , so weit es die Sensorik zulässt, $\text{cred}(c)$ die Information, wie sehr dieser Beschreibung zu trauen ist. Damit lassen sich auf der Weltmodellierung basierende Entscheidungen mit unterschiedlich hohem Risiko treffen. Um auf das Beispiel der in den Raum ragenden Tischkante zurückzukommen: $\text{occ}(c) = 0,5$ und $\text{cred}(c) = 0,9$ würde in diesem Fall mit großer Sicherheit uneinheitliches Terrain anzeigen, das man besser nicht betritt. Werte von $\text{occ}(c) = 0,9$ und $\text{cred}(c) = 0,5$ deuten auf eine einheitliche Landschaft hin. Im zweiten Fall wäre die Konsequenz, durch zusätzliche Messungen mehr Information zu akquirieren, im ersten Fall wäre das nicht sinnvoll.

Zusätzlich zu den genannten Datenfeldern ist es sinnvoll, eine Reihe von *Metadaten* in jeder Zelle zu halten, die zusätzliche Informationen über die gespeicherten unmittelbaren Daten liefern. Für Metadaten hat eine Zelle folgende Felder:

Anzahl vorheriger Messungen: Zunächst wird gespeichert, wie oft sich eine Messung auf diese Zelle ausgewirkt hat. Dies ist ein wichtiges Indiz für die Zuverlässigkeit der gespeicherten Werte. Da die Anzahl der Messungen monoton steigend ist, wäre zu überlegen, ob dieser Wert nicht ggf. zurückgesetzt werden sollte. Eine Möglichkeit wäre, dies zu tun, wenn die letzte Messung eine gewisse Altersgrenze überschreitet.

Alter der letzten Messung: Die Aktualität der in der Zelle gehaltenen Daten ist ein wichtiger Indikator, wie stark man sich auf diese verlassen kann. Dazu wird bei jedem Schreibzugriff auf die Zelle, auch wenn sie nicht modifiziert werden sollte, ein aktueller Zeitwert gespeichert. Der Aussagewert dieser Altersangabe ist allerdings leicht eingeschränkt, da nur das Alter der letzten Messung ablesbar ist, die Information der Zelle aber aus sehr vielen, eventuell sehr viel älteren Messungen gebildet wurde. Will man die Auswirkungen dieser Einschränkung verringern, kann man statt eines Zeitwertes eine Reihe der letzten n Messungen speichern und dann deren arithmetisches Mittel betrachten. Für kleine n lohnt dies allerdings nicht, da üblicherweise mehrere kurz aufeinander folgende Messungen ein und dieselbe Zelle betreffen, ein zu großer Wert n erhöht den Speicherverbrauch der Datenstruktur. Auch der Ansatz der so genannten *Temporal Occupancy Grids* (Arbuckle et al., 2002), der die Speicherung von Belegtheitsinformationen in verschiedenen Zeitskalen vorsieht, hat den Nachteil eines hohen Speicherverbrauchs. In dieser Arbeit werden daher ausschließlich die Zeitwerte des letzten Zugriffs gespeichert und bei der Bewertung des Alters die Anzahl der vorherigen Messwerte berücksichtigt.

Sensordichte: Die Sensordichte gibt an, wie viele verschiedene Informationsquellen an der in der Zelle gespeicherten Information beteiligt waren. Diese Quellen sind in erster Linie Sensoren, können aber auch mittelbare andere Informationsquellen sein. Beispielsweise ist das Wissen, dass an der gegenwärtigen Position des Roboterkörpers kein Hindernis sein kann, eine sehr verlässliche Information, die nicht direkt aus der Sensorik abgeleitet wird.

4.7.3 Einfügen neuer Sensormessungen

Wenn neue Sensordaten vorliegen, müssen diese zur Aktualisierung des bestehenden Weltmodells integriert werden. Es stellt sich die Frage, wie man vorgeht, um die in der betreffenden Zelle des Belegtheitsgitters gespeicherten Daten auf Grund der neu gewonnenen Informationen zu aktualisieren. Die alten Daten sollen dabei nicht einfach überschrieben werden.

Eine Möglichkeit, die beispielsweise auch in Murphy et al. (2002) verwendet wird, ist den Einfluss alter Daten exponentiell mit dem Alter abnehmen zu lassen. Dies berücksichtigt jedoch mit der Zeit nur einen einzigen Einflussparameter. Kruse et al. (1995) führen eine allgemeinere Gütefunktion ein, die den Einfluss der Messungen auf Grund der angenommenen Qualität der Messung ermittelt. Im Folgenden wird eine Funktion definiert, die Parameter für die Aktualisierung der Zelldaten liefert. Dabei soll das

gesamte Wissen einbezogen werden, das sowohl über die bereits gespeicherten Daten als auch über die aktuelle Messung an Rahmenparametern vorliegt. Auf Grund dieser Gütefunktion soll dann der Einfluss der neu eingehenden Daten gewichtet werden.

Weitere Einflussgrößen, die für die Bewertung der alten und neuen Daten herangezogen werden können, sind:

Sensorqualität: Die Sensorqualität ist ein Wert zwischen 0 und 1, der eine generelle Aussage darüber trifft, wie sehr den Daten des aktuell messenden Sensors zu trauen ist. Diese Bewertung ist Expertenwissen und muss einmalig extern vorgegeben werden.

Nachbarschaftsbeziehung: Bei der Betrachtung der Nachbarschaftsbeziehung zwischen den Gitterzellen geht man von der Annahme aus, dass Hindernisse meist zusammenhängend sind und einzelne Punkte auf Fehlmessungen, Rauschen oder Artefakte zurückzuführen sind. Eine Messung einer belegten Zelle in der Nachbarschaft anderer belegter Zellen lässt im Normalfall nicht auf einen Messfehler schließen. Es ist daher sinnvoll, ein Maß für den Belegtheitsstatus der unmittelbaren Nachbarschaft der Zelle als zusätzliche Einflussgröße zu betrachten.

Sei D_c die Menge aller Zellen in der Nachbarschaft von einer Zelle c , für die schon Beobachtungen vorliegen. Dann ist für einen für c beobachtbaren Belegtheitswert $s \in \{0, 1\}$

$$n(c, s) = \frac{\sum_{b \in D_c} |d(c, b)(s - \text{occ}(b) \text{cred}(b))|}{|D_c|}$$

ein Maß für die Ähnlichkeit der Nachbarschaft von c . $d(c, b)$ ist hierbei ein Gewichtungsfaktor, für den sich bei größer gewählten Nachbarschaften die Entfernung von b zu c anbietet, um nähere Nachbarzellen stärker zu gewichten. Im Folgenden wird nur die unmittelbar angrenzenden Zellen betrachtet und als Abstandsmaß $d(c, b) \equiv 1$ verwendet.

Bewegungsstatus des Roboters: Es ist beobachtbar, dass es vor allem dann zu ungenauen Messungen kommt, wenn der Roboter sich bewegt. Das liegt vor allem daran, dass einige Sensorwerte einen Median- und Durchschnitts-Filter durchlaufen. Durch deren integrativen Effekt reagieren die Sensoren auf Veränderungen durch Positionsverlagerungen langsamer als die Sensordaten der Gelenke, die diese Filter nicht durchlaufen. Im Ruhezustand der Maschine sind daher genauere Sensorwerte zu erwarten.

Zusammenfassend werden als Eingangsgrößen der Gütefunktion folgende Daten herangezogen: Zuverlässigkeit der bestehenden Daten, Alter der letzten Messung, Anzahl vorheriger Messungen, Sensordichte, Sensorqualität, Nachbarschaftsbeziehung und Bewegungsstatus des Roboters. Mit Hilfe dieser Eingangsgrößen soll eine geeignete Gütefunktion (siehe Abschnitt 4.7.4) zwei Werte als Ausgabe berechnen:

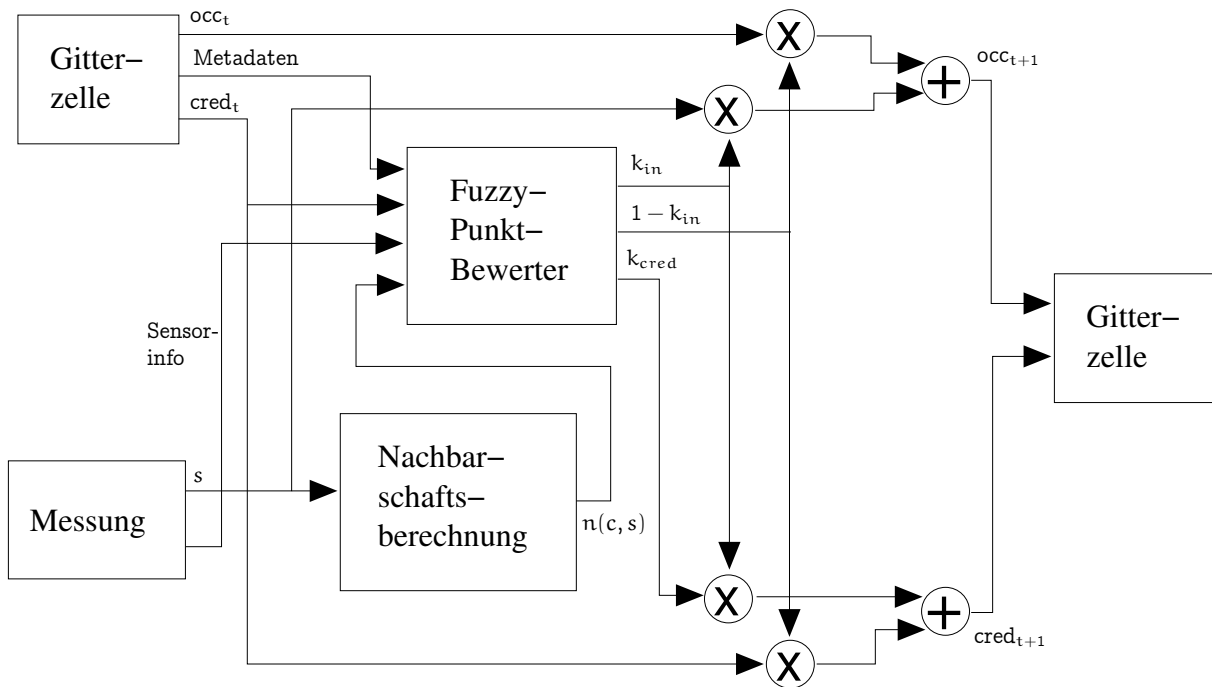


Abbildung 4.7: Schematische Darstellung des Datenflusses bei der Berechnung der neuen Werte einer Gitterzelle

- Einen *Zuverlässigkeitsparameter* k_{cred} , der eine Aussage über die Verlässlichkeit der aktuellen Messung liefert. Zu dessen Berechnung werden keine Daten aus dem bisherigen Datenbestand der Zelle herangezogen.
- Einen *Einflussparameter* k_{in} , der dazu dient, zu bestimmen, wie stark die aktuelle Messung (und auch die gleichzeitig geschätzte Zuverlässigkeit) in den Datenbestand eingehen soll.

Die zu betrachtende Zelle c hält zu einem Zeitpunkt t die Werte $\text{cred}_t(c)$ und $\text{occ}_t(c)$. Eine neue Sensormessung zum Zeitpunkt $t + 1$ liefert für die Zelle einen neuen Belegtheitszustand $s \in \{0, 1\}$ – entsprechend, ob ein Hindernis erkannt wurde oder nicht. Mit den aus der Gütefunktion gewonnenen Parametern k_{cred} und k_{in} aktualisiert sich der Zellinhalt nun wie folgt:

$$\begin{aligned}\text{occ}_{t+1}(c) &= (1 - k_{\text{in}}) \text{occ}_t(c) + k_{\text{in}} s \\ \text{cred}_{t+1}(c) &= (1 - k_{\text{in}}) \text{cred}_t(c) + k_{\text{in}} k_{\text{cred}}\end{aligned}$$

Abbildung 4.7 stellt die Ein- und Ausgänge und ihre gegenseitige Einflussnahme noch einmal schematisch dar. Für die Realisierung der Gütefunktion wurde eine Bewertung realisiert, die mit Methoden der Fuzzy-Logik arbeitet.

4.7.4 Realisierung der Gütefunktion

Die Verarbeitung einer großen Anzahl von Eingangsgrößen, wie sie dargestellt wurden, macht es schwierig, einen mathematischen Formalismus zu finden, der die gewünschten

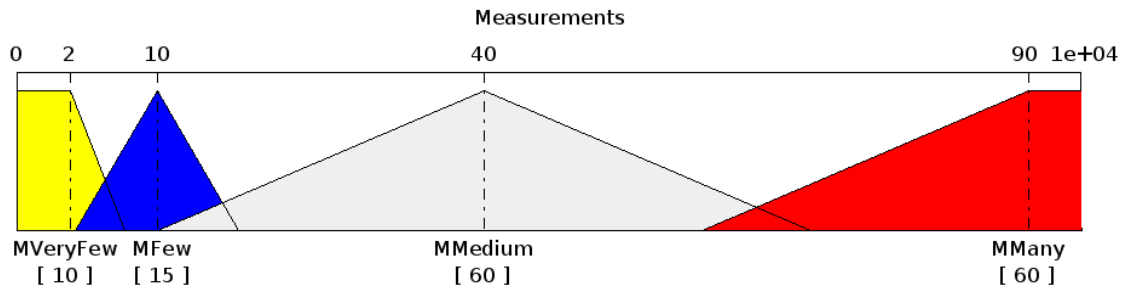


Abbildung 4.8: Fuzzy-Mengen für die Eingangsgröße „Anzahl der Messungen“. Die Größe der Mengen ist unterschiedlich, Messanzahlen im sehr geringen Bereich können so im Regelsatz gesondert berücksichtigt werden, hier fallen auch kleinere Unterschiede stark ins Gewicht (vergleiche Anhang C).

Verläufe beschreibt. Mit Hilfe von Methoden Unscharfer Logik ist es dagegen durchaus möglich, auf einfache Weise mathematisch nicht exakt fassbares Modellwissen zu verarbeiten.

An dieser Stelle sei angemerkt, dass Fuzzy-Methoden nicht mit der Exaktheit arbeiten, die eine geschlossene mathematische Form bieten würde. Es ist dem aber entgegenzuhalten, dass die Exaktheit im Modellwissen auch nicht existiert. Regeln wie „Wenn die bestehenden Daten sehr alt sind, dann gewichte die neuen relativ hoch“ beschreiben insofern recht genau die einfache dahinter stehende Heuristik. Eine mathematische Formel wäre im Bemühen um eine wirklichkeitsnahe Modellierung genauso willkürlich aus diesen wenig exakten Beobachtungen erstellt, es ist daher mehr als zweifelhaft, ob die Resultate, die sie liefert, der Realität eher gerecht werden. An dieser Stelle sei noch einmal auf die weiteren Ausführungen in Bezug auf Fuzzy-Logik im Anhang B sowie in Abschnitt 3.4.2 verwiesen.

Um nun zu einer Bewertung der Sensordaten zu kommen, werden für alle der genannten Eingangsgrößen auf ihrem jeweiligen Wertebereich mehrere Fuzzy-Mengen definiert, die qualitativen Bewertungen der Größe entsprechen. Für das Alter werden dies beispielsweise Terme wie „sehr alt“, „alt“ und „neu“ sein, für die Sensorqualität „hoch“ oder „gering“. Als Beispiel findet sich der gesamte verwendete Satz von Fuzzy-Mengen für die Eingangsgröße „Anzahl der Messungen“ in Abbildung 4.8.

Genauso werden auch für die beiden Ausgangsgrößen „Einfluss“ und „Zuverlässigkeit“ Fuzzy-Mengen definiert. Danach können Implikationsregeln in der Form

WENN Nachbarschaft = mittel	DANN $k_{\text{cred}} = \text{hoch}$
WENN Sensorqualität = hoch	DANN $k_{\text{in}} = \text{hoch}$
WENN Messanzahl = viel UND Alter = veraltet	DANN $k_{\text{in}} = \text{sehr hoch}$

aufgestellt werden. Für jede Sensormessung wird nun ein Inferenzschritt durchgeführt und die Parameter k_{in} und k_{cred} ermittelt. Abbildung 4.9 zeigt an zwei Beispielen das Ergebnis der Inferenz: Es sind die Werte der beiden Ausgabeparameter aufgetragen, wenn man jeweils eine Eingabegröße (hier Alter der letzten Messung und Anzahl der Messungen) ihren gesamten Definitionsraum durchlaufen lässt. Der komplette verwendete Regelsatz findet sich im Anhang C.2.

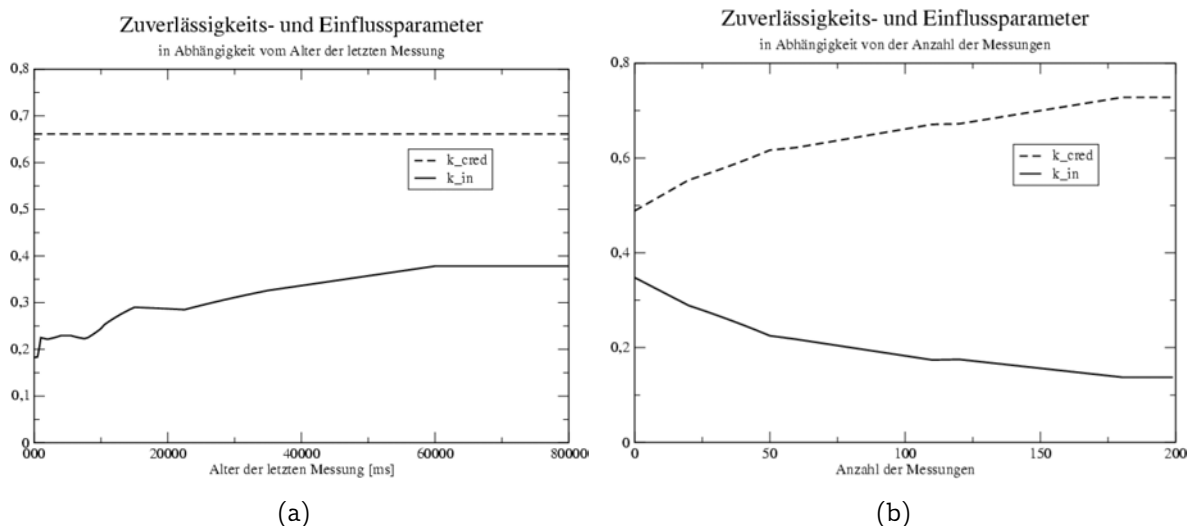


Abbildung 4.9: Der Verlauf der Ausgabeparameter bei unterschiedlichen Eingabewerten (a) Alter der letzten Messung (b) Anzahl der Messungen. Für beide Diagramme wurden bis auf eine alle Eingangsgrößen konstant gehalten und eine ihren gesamten Definitionsbereich durchlaufen lassen. In Beispiel (a) ist k_{cred} konstant, da dieser Parameter nicht vom Alter der letzten Messung abhängt.

4.8 Globale 2 $\frac{1}{2}$ D-Karten

Die im letzten Abschnitt vorgestellte dreidimensionale Urkarte dient in erster Linie zur Sensordatenfusion im Nahfeld. Um die Wegbeschaffenheitsinformation längerfristig zur Verfügung zu stellen, werden weitergehende zweidimensionale Karten benötigt, welche die Höheninformation des Geländes speichern. Da diese das Wissen bezüglich der 3. Dimension repräsentieren, spricht man von 2 $\frac{1}{2}$ D-Karten.

Zur Generierung eines 2 $\frac{1}{2}$ D-Modells aus einem 3D-Modell muss aus der Menge aller übereinanderliegenden Gitterzellen des Urmodells ein geeigneter Repräsentant ausgewählt werden. Die einfachste Methode stellt die Projektion von oben nach unten dar, führt aber nicht immer zu zufriedenstellenden Ergebnissen:

Beispiel 4.2 *Eine Laufmaschine stehe vor zwei Bäumen, deren Äste dazwischen eine geschlossene Decke bilden. Im Gegensatz zum 3D-Modell ist der Durchgang unter den Ästen in einem 2 $\frac{1}{2}$ D-Modell nicht erkennbar. Die Höheninformation der Äste überdecken die Höheninformation des Untergrunds. Ein Weg zwischen den Bäumen hindurch kann nicht gefunden werden.*

Diesem Effekt der Überdeckung des Untergrunds soll deshalb durch geeignete Maßnahmen entgegengewirkt werden: Sind in der 3D-Karte mehrere Ebenen identifizierbar, auf denen sich die Laufmaschine bewegen könnte, soll diejenige als Repräsentant ausgewählt werden, die der aktuellen Position der Laufmaschine am nächsten ist.

Weiterhin stellt sich noch die Frage nach dem Zeitpunkt der Generierung der 2 $\frac{1}{2}$ D-Karte. Eine Möglichkeit bestünde darin, die Karte bei Bedarf zu aktualisieren, also beim Verschieben der Urkarte und vor Beginn eines Planungsvorgangs. Dabei müsste jedoch die gesamte Urkarte Zelle für Zelle abgearbeitet werden, um die oben beschriebenen

Ebenen zu extrahieren. Dieser Aufwand kann vermieden werden, indem die Aktualisierung der $2\frac{1}{2}$ D-Karte mit dem Eintragen eines neuen Datums in die Urkarte durchgeführt wird. Dies hat einerseits den Vorteil, dass die $2\frac{1}{2}$ D-Karte immer auf dem aktuellen Stand ist. Andererseits ist nicht jede Veränderung einer Zelle in der Urkarte für die $2\frac{1}{2}$ D-Karte auch relevant. Kleinere Änderungen am Wert der Belegheitswahrscheinlichkeit bzw. der Zuverlässigkeit müssen nicht sofort in die Höhenkarte übernommen werden. Somit entfallen für einen Großteil der Messungen die weiteren Betrachtungen (man denke nur an die vielen Freiraumzellen zwischen Abstandssensor und Auftreffpunkt).

Die folgenden Ausführungen skizzieren den schrittweise Aufbau der $2\frac{1}{2}$ D-Karte. Es finde eine relevante Änderung der Gitterzelle $c^{3D} = (x^{3D}, y^{3D}, z^{3D})^T$ in der Urkarte statt. Diese wird bei der Generierung der $2\frac{1}{2}$ D-Karte nur dann als belegt angenommen, wenn sowohl die Belegheitswahrscheinlichkeit $\text{occ}(c^{3D})$ dieser Zelle über einer unteren Schranke occ_{\min} liegt als auch die Zuverlässigkeit einen Schwellwert (cred_{\min}) übersteigt. Die korrespondierende Zelle der $2\frac{1}{2}$ D-Karte sei $c^{2D} = (x^{2D}, y^{2D})^T$ mit zugehöriger Höheninformation h^{2D} . Ist die Gitterzelle c^{2D} noch undefiniert, fand bisher noch keine Messung an dieser Stelle statt, so dass die Höhe von c^{3D} direkt übernommen werden kann: $h^{2D} = z^{3D}$. Liegt andererseits schon eine Höheninformation h^{2D} vor, muss geklärt werden, welche Höhe aus der Urkarte in die $2\frac{1}{2}$ D-Karte übernommen werden soll. Hierbei spielt die zum Laufen benötigte „Kopffreiheit“ f_h eine wichtige Rolle: Ein in der $2\frac{1}{2}$ D-Karte verzeichneter Eintrag muss diesen Freiraum über der eingetragenen Höhe h^{2D} garantieren. Zur Berechnung des neuen Höhenwertes findet eine weitere Fallunterscheidung statt:

$z^{3D} > h^{2D}$ und $z^{3D} \leq h^{2D} + f_h$: Die neue Messung fällt in den Freiraum über der bisher eingetragenen Höhe, so dass der alte Eintrag in der $2\frac{1}{2}$ D-Karte ungültig wird. Ausgehend von der Messhöhe z^{3D} muss eine Zelle in der Urkarte mit geeignetem großem Freiraum gesucht werden.

$z^{3D} > h^{2D} + f_h$: Die neue Messung liegt über dem Freiraum, so dass der aktuelle Eintrag h^{2D} grundsätzlich weiterhin gültig bleibt. Ist über der neuen Messung jedoch ebenfalls genügend Freiraum vorhanden, muss entschieden werden, welche der beiden Möglichkeiten vorzuziehen ist. Der Höhenwert mit dem geringstem Abstand zur aktuellen Roboterhöhe ist hierfür der geeignetste, wenn man bedenkt, dass die Urkarte nur die unmittelbare Umgebung abbildet und der Roboter seine Höhe in der Regel nicht sprunghaft ändert.

$z^{3D} \leq h^{2D}$: Die neue Messung liegt unterhalb des aktuellen Eintrags, so dass der aktuelle Eintrag gültig bleibt. Verfügt die Messung selbst ebenfalls über genügend Freiraum f_h , wird wie im letzten Fall beschrieben zwischen beiden ausgewählt.

Liegt die Belegheitswahrscheinlichkeit der geänderten Urkarten-Zelle c^{3D} unter der minimalen Belegtheit $\text{occ}(c^{3D}) \leq \text{occ}_{\min}$ oder ist die Glaubwürdigkeit der Messung zu gering $\text{cred}(c^{3D}) \leq \text{cred}_{\min}$, gilt diese bei der Generierung der $2\frac{1}{2}$ D-Karte als frei. Wird eine bisher da gewesene Zelle in der Urkarte entfernt, muss unter Umständen die Höheninformation in der $2\frac{1}{2}$ D-Karte aktualisiert werden. War bisher genau die Messhöhe

verzeichnet ($h^{2D} = z^{3D}$), wird die Eintragung h^{2D} ungültig. Es muss unter allen möglichen Kandidaten der Urkarte mit genügend Freiraum ein geeigneter ausgewählt werden. Ansonsten kann die frei gewordene Zelle einer tieferliegenden Zelle der Urkarte zu einem gültigen Freiraum verholfen haben, so dass wiederum zwischen dieser und dem aktuellen Höheneintrag gewählt werden muss.

4.9 Zusammenfassung

Zur Unterstützung des Planungsprozesses ist eine rechnerinterne Repräsentation der realen Welt erforderlich. Durch Fusion aller zur Verfügung stehenden Sensordaten soll daher während des Laufens eine Karte der Umgebung generiert werden. Nach wie vor besteht die Forderung nach einfacher, rechnereffizienter Sensorik. Detaillierte a-priori-Karten oder strukturierte Umweltmerkmale (natürliche Landmarken) wie beispielsweise Kanten können in unstrukturiertem Gelände nicht erwartet werden. Darüber hinaus sollte das Umweltmodell robust gegenüber Lokalisationsfehlern sein und Hindernisse möglichst exakt wiedergeben können. Zur Planung der Fußaufsetzpunkte benötigen Laufmaschinen insbesondere im Nahbereich einen hohen Detaillierungsgrad. Die Möglichkeit zum gezielten Nachmessen sowie zur Extraktion metrischer Information sollte gegeben sein.

Zufrieden stellende topologische Karten können unter den gegebenen Rahmenbedingungen nur schwer aufgebaut werden; die geforderte detaillierte, metrische Information lässt sich nur mit hohem Aufwand gewinnen. Geometrische Ansätze, vorzugsweise in Form eines Belegtheitsgitters, werden den in dieser Arbeit gestellten Anforderungen besser gerecht. Es hat sich herausgestellt, dass sich eine lokale, dreidimensionale Karte in Kombination mit einer gleichzeitig aufzubauenden globalen $2\frac{1}{2}D$ -Karte zur Umweltmodellierung von Laufmaschinen im Gelände am ehesten eignet.

Die Sensordatenfusion findet im entwickelten *erweiterten Inferenz-Gitter* unter Verwendung unscharfer Regeln in 3D statt. Die einzelnen Gitterzellen speichern neben einer Belegtheitswahrscheinlichkeit einen zusätzlichen Zuverlässigkeitsparameter und weitere Metadaten wie beispielsweise Alter der letzten Messung, Anzahl vorheriger Messungen oder die bisherige Sensordichte. Neben weiteren Eigenheiten aktueller Messungen wie Sensorqualität, Bewegungsstatus des Roboters und Nachbarschaftsbeziehung dienen die gespeicherten Daten der Fuzzy-Bewertung als Eingangsgrößen zur Aktualisierung der Zelleninhalte.

Für weitergehende Planungsaufgaben wird eine globale zweidimensionale Höhenkarte bereitgestellt. Dieses $2\frac{1}{2}D$ -Modell wird ständig aktuell gehalten. Mit Hilfe spezieller Methoden werden Mehrdeutigkeiten im Dreidimensionalen möglichst durch die für die Laufmaschine geeignetste Höheninformation wiedergegeben.

Kapitel 5

Bahnplanung

Planung ist die gedankliche Vorwegnahme zukünftigen Handelns durch Abwägen verschiedener Handlungsalternativen und Entscheidungen für den günstigsten Weg.

(Wöhe, 2000)

Diese auf den denkenden Menschen bezogene Definition von *Planung* lässt das Planungsziel grundsätzlich offen. Ist der zu erreichende Zielort eines mobilen Systems hingegen vorgegeben, dient Planung der Generierung einer möglichst optimalen Abfolge zukünftiger Aktionen zur Überführung des Robotersystems aus der gegebenen Ausgangskonfiguration in die definierte Zielkonfiguration. Bezogen auf die Navigation erstreckt sich die Planung im Wesentlichen auf die *Bahnplanung*: Ausgehend von einer Startposition und -orientierung im Raum (im Allgemeinen die aktuelle Position und Orientierung) ist ein Weg durch den Raum hin zu einer Zielposition zu planen, welcher Hindernisse entsprechend den Möglichkeiten bzw. Bewegungsfreiheitsgraden des Roboters umgeht. Im Folgenden wird nun erörtert, wie dieser Planungsvorgang für Laufmaschinen gestaltet werden kann. Im Rahmen von Studien- und Diplomarbeiten (Huber, 2005) wurden hierzu wertvolle Beiträge geleistet.

5.1 Anforderungen und Randbedingungen

Je nach Abstraktionsgrad fließen die spezifischen Fähigkeiten eines mobilen Systems unterschiedlich stark in die Bahnplanung mit ein. Bei einer *globalen Bahnplanung* wird auf die Unterscheidung verschiedener Antriebsarten verzichtet. Das hohe Abstraktionsniveau erlaubt die Planung eines abstrakten Wegs wie er etwa auf Landkarten oder Stadtplänen zu finden ist. Eine derartige Vorgehensweise ist nicht Gegenstand der vorliegenden Arbeit; es wird vorausgesetzt, dass ein globaler applikationsspezifischer Plan, bestehend aus einer Abfolge von Zielpunkten, durch einen Leitstand bereitgestellt wird.

Die folgenden Ausführungen fokussieren sich auf die Generierung der eigentlichen Bahn, welche einen kollisionsfreien Weg des Zentralkörpers beschreibt und die vorgegebenen Zwischen- bzw. Zielpunkte miteinander verbindet. Bei der Wegsuche handelt es sich um ein Planungsproblem, dessen Komplexität mit der Anzahl der Bewegungsfreiheitsgrade des mobilen Roboters und der Anzahl der Randbedingungen skaliert (Dose,

1995). Randbedingungen sind dabei:

Hindernisse: Die im Umweltmodell gespeicherte Information über mögliche Hindernisse. Diese bildet die Grundlage für Entscheidungen, wie ein Hindernis zu umgehen ist.

Korrekturmöglichkeiten: Abhängig vom Einsatzgebiet des Roboters ist es notwendig, eine bereits geplante Bahn zu verändern, weil z.B. Hindernisse sensorisch nicht oder nur ungenau erfasst wurden, dynamische Hindernisse die Bahn kreuzen oder eine Verfolgung der Bahn fehlerhaft ist.

Optimalitätskriterien: Die Planung soll optimale Ergebnisse liefern, beispielsweise hinsichtlich des kürzesten Wegs.

Des Weiteren werden an die Bahnplanung noch folgende Anforderungen gestellt:

Rechenaufwand: Die Planung soll vor Ort durch die Laufmaschine selbst durchgeführt werden. Somit ist auch der Rechenaufwand zu berücksichtigen. Können die Berechnungen des Planers nicht mit der Hardware des Roboters in hinreichender Zeit durchgeführt werden, ist das Verfahren nicht geeignet.

Vollständigkeit: Ein Bahnplaner heißt vollständig, wenn er genau dann einen Weg liefert, falls ein Weg existiert (Dose, 1995).

5.1.1 Besondere Anforderungen in unstrukturiertem Gelände

Der Einsatz in unstrukturiertem, natürlichem Gelände setzt zusätzliche, einschränkende Rahmenbedingungen:

Komplexe Hindernisgeometrien: Unstrukturiertes Gelände bringt die unterschiedlichsten Arten an Hindernissen mit sich. Diese lassen sich selten durch einfache Geometrien hinreichend beschreiben. Dies wiederum hat zur Folge, dass sich Bahnplanungsalgorithmen, die sich direkt an Hindernisgeometrien orientieren, nicht einsetzbar sind. Hierzu sind u.a. die Wegenetzverfahren (z.B. Sichtbarkeitsgraph) zu zählen, deren Zeitaufwand mit $O(n^2)$ skaliert, wobei n die Anzahl der Kanten aller Hindernispolygone ist (Latombe, 1991).

Keine Begrenzungsflächen: Natürliches Gelände weist in den seltensten Fällen definierte Begrenzungsflächen wie beispielsweise die Wände eines Gebäudes auf. Zumindest kann nicht davon ausgegangen werden, dass eine räumliche Begrenzung des Suchraumes a-priori vorgegeben ist.

Unbekanntes Terrain: Da initial keine fein granulierte Karte der Umgebung bekannt ist, muss die Wegsuche die Behandlung unbekannter Gebiete erlauben.

5.1.2 Zusätzliche Anforderungen bei Laufmaschinen

In Hinblick auf die Bahnplanung stellt die Fähigkeit von Laufmaschinen, Hindernisse nicht nur umgehen, sondern auch überwinden zu können, wohl die größte Herausforderung dar.

Überwindbare Hindernisse: Der Planungsalgorithmus muss einen kritischen Umgang mit dem Begriff „Hindernis“ erlauben, was die Komplexität des Planungsproblems steigert.

Einfache Wege: Laufmaschinen sind zwar durchaus befähigt, Hindernisse zu übersteigen, jedoch ist es einleuchtend, dass das Laufen auf ebenem Boden ungleich einfacher, sicherer und somit in der Regel auch schneller auszuführen ist. Dies sollte sich in einem Optimalitätskriterium und somit auch im geplanten Weg niederschlagen, womit eine erneute Zunahme der Komplexität verbunden ist.

5.1.3 Das Umweltmodell als Planungsgrundlage

Wird neben der Umwelt auch der Roboter in die Modellierung mit einbezogen, so spricht man von einer *Modellierung des Konfigurationsraums*, bei der die Bewegungsfreiheitsgrade des Roboters den Koordinatenachsen des Raumes entsprechen (Spandl, 1991). Hierdurch würde zwar die Planung vereinfacht, aber die Berechnung des Raumes ist gerade bei einer höheren Anzahl an Bewegungsfreiheitsgraden äußerst komplex (Lozano-Pérez, 1983). Auf Grund der Tatsache, dass eine Laufmaschine wegen ihrer Beine grundsätzlich viele Bewegungsfreiheitsgrade besitzt, ist die Berechnung des Konfigurationsraums nicht praktikabel.

Mit der Festlegung des Umweltmodells in Kapitel 4 auf ein geometrisches Belegheitsgitter muss sich auch das zu entwickelnde Planungsverfahren an Geometrien orientieren. Soll von den speziellen Fähigkeiten der Laufmaschinen zur Hindernisüberwindung nicht vollständig abstrahiert werden, sind Höheninformationen des Untergrunds in der Modellierung unabdingbar. Bleibt noch offen, in welcher Ausprägung das Umweltmodell für die Planung herangezogen werden soll: 3D oder $2\frac{1}{2}D$? Die dreidimensionale Urkarte steht nur in der unmittelbaren Umgebung der Laufmaschine zu Verfügung. Eine weitläufige Bahnplanung kann somit nur auf Basis der globalen $2\frac{1}{2}D$ -Karte stattfinden.

Bei der Verwendung eines $2\frac{1}{2}D$ -Modells wird die Reduktion der Planungskomplexität durch eine Reduktion der möglichen Wege erkauft. Dies kann zur Folge haben, dass die Planung keine Lösung liefert, obwohl in Wirklichkeit ein Weg existiert. Trotzdem erscheint der Kompromiss einer $2\frac{1}{2}D$ -Karte tragfähig: Schließlich werden keine Wege ermöglicht, die tatsächlich nicht vorhanden sind. Weiterhin fließt das Vorwissen um übereinander liegende 3D-Punkte beim Projektionsvorgang mit ein. Über Kopf hängende Strukturen, die für den Roboter kein Hindernis darstellen, werden ignoriert und der Verlauf des betretbaren Untergrunds wiedergeben. Jedoch muss auch ein derartiges Projektionsverfahren beispielsweise entscheiden, welcher der beiden real nutzbaren Wege einer Brücke schließlich abgebildet werden soll; beide Wege sind in einer $2\frac{1}{2}D$ -Karte nicht modellierbar.

5.2 Planungsverfahren

Zur Lösung der Planungsproblematik, insbesondere der Wegesuche radgetriebener Roboter, existiert eine Vielzahl von Veröffentlichungen (Latombe, 1991). In diesem Abschnitt werden einige Verfahren exemplarisch erläutert, um einen Überblick über die verschiedenen vorgeschlagenen Lösungen zu ermöglichen. Insbesondere soll dargestellt werden, auf welcher Grundlage die Entscheidungen für den in dieser Arbeit realisierten Ansatz getroffen wurden.

5.2.1 Distance Transform (Entfernungstransformation)

Bahnplanungsalgorithmen, basierend auf Distance Transform (DT), liefern einen optimalen kollisionsfreien Weg (Jarvis, 1994). Für eine bekannte, stationäre Karte der Umgebung und einem Zielpunkt liefert das Verfahren ein zweidimensionales Array von Zellen. Jede dieser Zellen enthält einen Wert, welcher die Kosten einer kollisionsfreien Distanz zum Ziel hin repräsentiert. Initial enthält die Zielzelle den Wert 0 als Distanzkosten, während alle hindernisfreien Zellen einen sehr hohen Wert erhalten. Die Distanzkosten jeder Zelle werden dann durch Propagieren der Kosten benachbarter Zellen bestimmt. Hinderniszellen werden von den Berechnungen ausgeschlossen. Am Ende gibt der Wert einer Zelle an, wieviel Schritte bzw. Zellen wenigstens benötigt werden, um das Ziel zu erreichen.

Das Verfahren ist zwar recht einfach, liefert in bekannten und geschlossenen Umgebungen aber sehr gute Ergebnisse. In der freien Natur eignet es sich dagegen wenig. Eine Erweiterung auf bekannte, offene Umgebungen wird von Chin et al. (2001) vorgestellt. Offen bleiben aber weiterhin die Behandlung von unbekanntem Gelände und die Erweiterung auf Laufmaschinen, da die Entscheidung der Hindernisart nur für Radfahrzeuge getroffen wird. Das Speichern von Kosten der Zellen wird allerdings in der vorliegenden Arbeit übernommen, um den späteren Berechnungsaufwand zu reduzieren.

5.2.2 Dekompositionsverfahren

Bei Dekompositionsverfahren wird der gesamte Raum als Graph repräsentiert. Die auf Gitterkarten operierenden Varianten definieren einen Graph, dessen Knoten belegten oder freien Gitterelementen entsprechen. Benachbarte Gitterelemente besitzen dabei gemeinsame Vorgängerknoten. Dadurch lässt sich die Bahnplanung als Graphsuche gestalten (Abbildung 5.1). Die Wegplanung selbst hat einen Aufwand von $O(n \log n)$, wobei n die Anzahl der Gitterelemente bezeichnet (Latombe, 1991).

Die Erweiterung dieser Art Verfahren für den Einsatz auf Laufmaschinen würde eine flexiblere Definition von Hindernissen erfordern. Gitterelemente müssten im Graph abhängig von den Vorgängerknoten mehrfach repräsentiert sein: als frei, wenn das Betreten vom Vorgängerknoten aus möglich ist bzw. als belegt, wenn dem nicht so ist. Der Aufwand zur Erzeugung des Graphen würde den Aufwand zur Wegplanung weit überschreiten. Der große Vorteil dieser Verfahren, die Umweltrepräsentation direkt in eine Graphsuche umzuformulieren, ginge verloren.

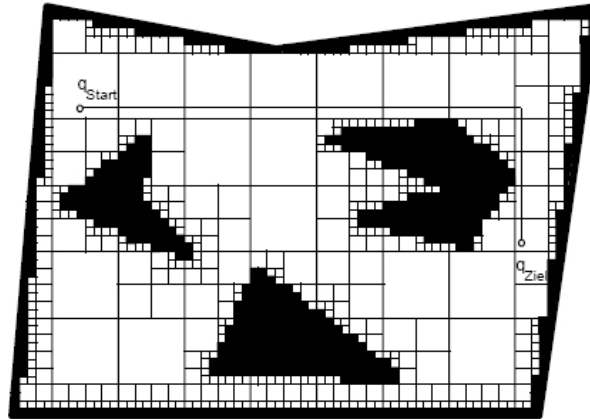


Abbildung 5.1: Dekompositionsverfahren: Hindernisse werden durch Gitterelemente approximiert. Die Feinheit der Approximation wird durch die Größe des kleinsten Gitterelementes, der Gitterzelle, bestimmt. Als Graph dient ein *Quadtree* (Viererbaum).

5.2.3 Lernverfahren

Um der Forderung nach Autonomie gerecht zu werden, bedarf es eines flexiblen Umgangs mit den Unsicherheiten, die in einer realen Umgebung stets vorhanden sind. Dies lässt sich durch Mechanismen erreichen, die den Roboter in die Lage versetzen, aus den im Laufe der Zeit gemachten Erfahrungen zu lernen und mit dem neu erworbenen Wissen eine Leistungssteigerung zu erzielen.

5.2.3.1 Erklärungsbasiertes Lernen

Zur Verbesserung der Bahnplanung einer vierrädrigen, mobilen Roboterplattform wird in Spandl (1991) ein erklärungsbasierter Lernansatz verfolgt. *Auftragsbezogene Makro-Trajektorien*, die durch Start- und Zielkonfiguration der Plattform gekennzeichnet sind, werden abgespeichert und erneut ausgewählt, wenn ein ähnlicher Fahrauftrag in das System eingeht. Sollte keine solche Trajektorie gefunden werden, erfolgt die eigentliche geometrische Bahnplanung. Bei dieser wird zunächst eine Grobplanung durchgeführt. Als Resultat ergibt sich eine Folge von Freiraumpolygonen. Innerhalb dieser erfolgt anschließend eine Feinplanung zur Sicherstellung kollisionsfreier Bahnen. Die Lösungen der Feinplanung werden generalisiert als *geometriebezogene Makro-Trajektorien* gespeichert und stehen so künftig ähnlichen Feinplanungen zur Verfügung. Zum Abschluss der geometrischen Wegsuche wird die gefundene kollisionsfreie Bahn als neue *auftragsbezogene Makro-Trajektorie* abgelegt.

Die Bildung von Makro-Trajektorien, oder allgemeiner Makro-Operatoren, löst allerdings noch nicht das eigentliche Planungsproblem. Für das gegebene Start-Ziel-Zustandspaar muss zunächst eine Lösung gefunden werden. Ein System, welches erklärungs-basiertes Lernen zur Findung eben solcher Lösungen verwendet, ist SOAR (Laird et al., 1986). Dieser Planer lernt aus dem Erklären von Situationen, in denen die gewählte Plan- oder Suchstrategie keinen Ausweg findet. Ist keine eindeutige Auswahl des nächsten Operators möglich, so verwendet SOAR einfache Mechanismen wie „Genera-

te-and-Test“, um die Sackgasse aufzulösen (Mitchell, 1997). Dadurch ist es dem System möglich, Informationen zum Finden eines Ausweges zu suchen. Dabei lässt sich der Vorgang zur Auflösung der gegenwärtigen Sackgasse als Erklärung für die Auflösung künftiger, ähnlicher Sackgassen auffassen. Eine Generalisierung, bezeichnet als *Chunking*, gibt die Bedingungen an, unter welchen die gefundene Erklärung anwendbar ist. Die so erhaltene Regel ermöglicht es SOAR, in Situationen eine Entscheidung zu treffen, die zuvor in einer Sackgasse gemündet hätten.

Die eigentliche geometrische Planung oder die Beurteilung der geometrischen Gegebenheiten wird durch erklärungs-basierte Lernverfahren nicht obsolet, sondern nur weniger häufig angewandt. Daher lösen diese Verfahren die Planungsaufgabe nicht, sondern stellen lediglich eine Verbesserung dar.

Die Verwendung von Makro-Operatoren zeigt ein generelles Problem des erklärungs-basierten Lernens auf, nämlich das der *Nützlichkeit* der gefundenen Regeln. So sollen die Makro-Operatoren eine Beschleunigung der Planungsaufgabe bewirken, da eventuell auf eine vorhandene Lösung zurückgegriffen werden kann. Paradoxerweise nimmt mit der Zeit die Effizienz des Systems ab. Es werden immer mehr Makros erzeugt, so dass die Suche nach einem geeigneten aufwändiger wird. Die Kosten für die Suche übersteigen den Nutzen.

5.2.3.2 Evolutionäre Algorithmen

Zur Lösung von Optimierungsaufgaben, zu denen auch das Finden eines Weges gehört, werden sehr häufig evolutionäre Algorithmen verwendet. Diese lehnen sich an das Evolutionsprinzip der Natur an. Mehrere Lösungshypothesen, die *Individuen*, die durch einen Merkmalsvektor repräsentiert werden, bilden eine Menge, die *Population*. Zur Bewertung der Güte einer Lösung dient eine Kosten- oder Fitness-Funktion, deren Ergebnis die Grundlage für die Fortpflanzung durch *Klonen*, *Mutieren* oder *Kreuzen* der Individuen bildet. Je höher die Fitness, umso höher die entsprechende Wahrscheinlichkeit für die Fortpflanzung bzw. das Überleben. Auf diese Weise kann der Lösungsraum nach der besten Lösung abgesucht werden.

Hocaoğlu und Sanderson (2001) setzen evolutionäre Algorithmen zur Lösung eines 2D-Bahnplanungsproblems ein. Abbildung 5.2 zeigt den Ablauf des Planungsfortschrittes. Eine Möglichkeit, die verschiedenen Individuen bzw. Wege auf ihre Korrektheit hin zu überprüfen, wäre die Ausführung einer Simulation. Optimalitätskriterien ließen sich durch geeignete Fitness-Funktionen bewerten.

Farritor und Dubowsky (1997) indizieren mögliche Roboteraktionen in einem so genannten *Action Module Inventory*. Ein Individuum bzw. *Aktionsplan* wird durch eine Folge von Aktionsindizes beschrieben. Um den Lösungsraum erkunden zu können, stehen Kreuzungs-Operatoren zur Verfügung, welche neue Aktionspläne aus bestehenden generieren. Zur Bewertung eines Planes wird dieser simuliert. Weitere Maße, wie etwa „aktuelle Distanz zum Ziel“ oder „Energieverbrauch“ lassen sich mit dem Simulationsergebnis zur endgültigen Fitness-Funktion verknüpfen. Durch die Zusammenfassung mehrerer häufig auftretender Aktionsfolgen zu einer Art Makro-Aktion wird der Roboter zum Lernen befähigt.

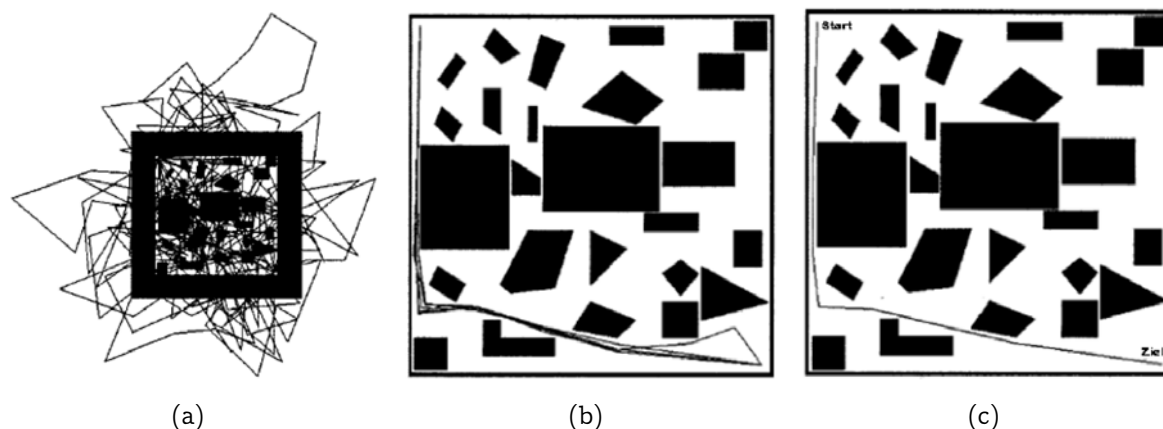


Abbildung 5.2: Anwendung evolutionärer Algorithmen auf ein 2D-Bahnplanungsproblem. (a) Initiale Population (b) nach 100 Generationen (c) der beste gefundene Weg zum Ziel. Aus (Hocaoğlu und Sanderson, 2001).

Grundsätzlich lässt sich feststellen, dass diese Ansätze vielversprechend sind, jedoch ein detailliertes Modell des Roboters und der Umwelt für die Simulation zur Verfügung stehen muss. Durch das Speichern und Bewerten vieler zahlreicher zu berücksichtigender Individuen sind diese Verfahren sehr rechenintensiv. Der Lösungserfolg hängt maßgeblich von einer geeigneten Wahl der Repräsentation des Problems ab (Mitchell, 1997). Ob ein optimaler Weg gefunden werden kann, ist nicht gesichert. Oft reicht eine gute Annäherung an das Optimum schon aus.

5.2.4 Potenzialfeldverfahren

Namensgebend für diese Methodik zur Bahnplanung sind virtuelle Potenzialfelder, welche Hindernissen und dem Ziel als implizite Abstandsfunktion zugeordnet werden. Meist werden mathematische Funktionen verwendet, bei denen ein Hindernis abstoßende und das zu erreichende Ziel anziehende Kräfte auf den Roboter ausüben (vergleiche Abbildung 5.3). Als Maß der virtuellen Kraftfelder können Potenzialfeldmodelle herangezogen werden. Jedem Raumpunkt kann somit ein Potenzial zugeordnet werden. Zur Wegeplanung können daher Gradientenabstiegsverfahren genutzt werden, da die Zielorte immer in den Minima des Potenzialfelds liegen (Khatib, 1986).

Problematisch bei der Gradientensuche ist das mögliche Vorhandensein lokaler Minima neben dem globalen Minimum des Zieles. Die lokalen Minima ergeben sich aus der Superposition der Potentiale mehrerer Hindernisse und können dazu führen, dass die Wegsuche bereits vorzeitig endet. Um dennoch das Ziel erreichen zu können, müssen weiterführende Verfahren angewendet werden, um das Verlassen oder Vermeiden lokaler Minima zu ermöglichen.

Üblicherweise werden bei den Potenzialfeldverfahren keine vollständigen Wege geplant, sondern immer nur lokale Bewegungen. Bei der Erfassung neuer Hindernisse ist daher ein Neuplanen des Weges nicht nötig. Lediglich die Potenzialfunktion muss neu berechnet werden. Die Wegeplanung kann online erfolgen, da kontinuierlich der Gradient an der aktuellen Position des Roboters zu berechnen ist. Insgesamt skaliert der

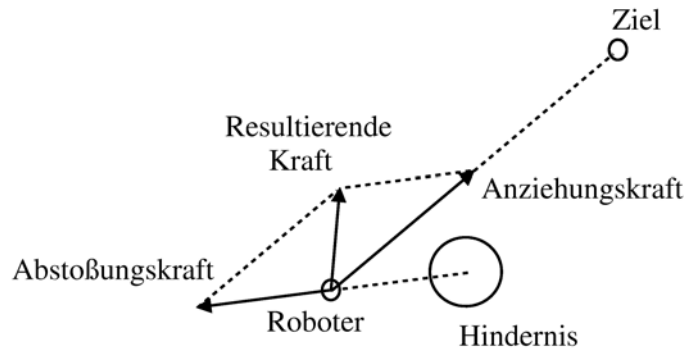


Abbildung 5.3: Potenzialfeldverfahren: Hindernisse üben Abstoßungskräfte, das Ziel eine Anziehungskraft auf den Roboter aus. Der Roboter folgt der resultierenden Kraft.

Aufwand mit der Anzahl der Hindernisse.

Dies sei an einem Beispiel erläutert: Um einen Raum, repräsentiert durch eine Gitterkarte, mit Hilfe eines Reinigungsroboters säubern zu können, wird in Prassler et al. (2000) vorgeschlagen, gleich zwei Potenzialfelder zu verwenden, um je nach Operationsmodus unterschiedliche Optimalitätskriterien zuzulassen. Hierbei wird der flexible Einsatz der Potenzialfeldverfahren deutlich. Dass eine Gitterkarte vorteilhaft bei der Berechnung der Potenziale sein kann, zeigt sich ebenso. So lassen sich Rechenaufwand reduzieren sowie Nachbarschaftsbeziehungen einfacher ausnutzen.

Mit mathematischen Funktionen zur Beschreibung der Potenziale kann eine Bahnplanung für Laufmaschinen jedoch nicht in dem Maße durchgeführt werden, wie es Anliegen dieser Arbeit ist. Es sollen Hindernisse schließlich nicht nur gemieden sondern auch überstiegen werden können. Daher ist eine realitätsnahe Modellierung der Hindernispotenziale erforderlich.

5.3 Erste Schlussfolgerungen

Die große Zahl der vorgestellten Verfahren lässt den Schluss zu, dass die Planung eines Weges insbesondere für radgetriebene Roboter auf Straßen oder in strukturierten Gebäuden einen fortgeschrittenen Entwicklungsstand aufweist. Eine direkte Übernahme unter Berücksichtigung der in Abschnitt 5.1 vorgestellten Anforderungen und Randbedingungen ist aus verschiedenen Gründen nicht möglich (Tabelle 5.1):

Interpretation der Hindernisse: Eine kategorische Unterscheidung in begehbar und nicht begehbar Bereiche geschieht oftmals mit zu einfachen Mitteln. Das bloße Heranziehen der Höhe eines Objekts wird der Flexibilität von Laufmaschinen nicht gerecht. Weitergehende Berechnungen sind für eine sinnvolle Unterscheidung notwendig. Gerade dies findet aber häufig nicht statt.

Rechenleistung: Sind mobile Plattformen mit genügend Traglast und Lastvolumen ausgestattet, können sie genügend Rechnerressourcen mitführen. In diesem Fall kann eine online-onboard Planung in Echtzeit erfolgen. Bei Laufmaschinen ist

	Distance Transform	Dekomposition	Erklärungsbasiertes Lernen	Evolutionäre Algorithmen	Potenzialfeld
Allgemein:					
Rechnereffizienz, Echtzeitfähigkeit	–	--	–	--	○
Vollständigkeit	○	○	○	○	○
Unstrukturiertes Gelände:					
komplexe Hindernisse	--	--	○	+	+
keine Begrenzungsflächen	+	--	○	○	+
unbekanntes Terrain	--	--	–	+	+
Laufmaschinen:					
Überwindbare Hindernisse	--	–	○	+	+
einfache Wege	--	–	○	+	++

++: gut geeignet +: geeignet ○: mäßig geeignet -: wenig geeignet --: ungeeignet

Tabelle 5.1: Beurteilung der untersuchten Planungsverfahren anhand der an sie gestellten Anforderungen: Weder Distance Transform noch die Dekompositionsverfahren können mit Unbekanntem oder komplexen, überwindbaren Hindernissen umgehen. Erklärungsbasiertes Lernen kann unterstützend wirken; die eigentliche Planungsaufgabe löst es aber nicht. Der Einsatz evolutionärer Algorithmen scheitert an deren unermesslichen Rechenaufwänden. Einzig die Potenzialfeldmethoden weisen genügend Flexibilität bei vertretbarem Rechenaufwand auf.

eine solche Ausstattung in der Regel nicht vorhanden, da sie klein, leicht und kompakt ausgelegt sind. Auch eine Verlagerung der Planung auf einen externen Rechner ist wegen der geforderten Autonomie nicht sinnvoll.

Aufwand: Lernverfahren erfordern eine geeignete Repräsentation, Trainingsbeispiele und Hintergrundwissen. Der Aufwand zur Verarbeitung von Trainings- und Lern-daten sowie die Lernalgorithmen sind komplex. Die gelernten Planungsverfahren bzw. Umgebungsmodelle sind dabei oft nur auf spezielle Begebenheiten zugeschnitten und nicht robust. Beides gilt es zu vermeiden.

Optimalität: Von allen möglichen Wegen soll der Beste gefunden werden. Dieses Ziel hängt stark von den gewählten Optimalitätskriterien ab. Planungsverfahren für radbasierte mobile Systeme sind wegen deren eingeschränkten Randbedingungen bezüglich Optimalität einfacher zu handhaben.

Unbekanntes Gelände: In vielen Überlegungen gilt die vereinfachende Annahme des vollständig bekannten Geländes. Dies mag für rein algorithmische Überlegungen ausreichend sein, ist es für den praktischen Einsatz aber nicht.

Dies sind die wesentlichsten Kritikpunkte der bisher vorgestellten Ansätze. Geeignete Ansätze für Laufmaschinen werden im nächsten Abschnitt aufgeführt und diskutiert.

5.4 Bahnplanung für Laufmaschinen

Wesentlich bei der Bahnplanung von Laufmaschinen ist eine Beurteilung, ob und wie die einzelnen Beine des Roboters an einer bestimmten Stelle platziert werden können. Legt man die vorliegende geometrische 2½D-Umweltmodellierung zugrunde, so lässt sich der Aufwand für jeden Planungsschritt wie folgt nach oben abschätzen (Bai und Low, 2001):

$$e(n, k) = \sum_{j=0}^{n-3} k^j \binom{n}{j} \quad \text{mit} \quad \begin{array}{l} n = \text{Beinanzahl} \\ k = \text{Platzierungsmöglichkeiten eines Beins} \end{array} \quad (5.1)$$

Diese Gleichung gibt an, wie viele Möglichkeiten bestehen, 0, 1, ... oder $n - 3$ Beine pro Planungsschritt zu platzieren. Drei Beine müssen letztlich auf dem Boden bleiben, um die statische Stabilität zu sichern. Es wird deutlich, dass eine möglichst exakte Planung sehr rechenintensiv ist.

5.4.1 Chen und Kumar (1996)

Ein suboptimaler Weg ist in vielen Fällen ausreichend, wenn dieser in Echtzeit bestimmt werden kann. Eine Idee, dies zu erreichen, wird in Chen und Kumar (1996) vorgestellt. Anstatt zu jedem Zeitpunkt jede Möglichkeit der Beinplatzierung durchzutesten, um so an das Ziel zu gelangen, wird eine bestimmte Anzahl von Beinplatzierungs- bzw. Fuß-aufsetzpunktfolgen zufällig ausgewählt. Die Zusammensetzung einer Schrittfolge steuert

eine Heuristik, um den Fortschritt der Planung zum Ziel hin zu gewährleisten. Unter allen gewählten Schrittfolgen wird dann die Beste ermittelt. Die Folgenanzahl richtet sich dabei nach einer zuvor festgelegten Wahrscheinlichkeit, mit welcher der resultierende Pfad nahe dem gesuchten Optimum sein soll.

Beispiel 5.1 Sei p die Wahrscheinlichkeit, dass der resultierende Pfad „gut“ ist und sei α die Wahrscheinlichkeit einer „guten“ Fußaufsetzpunktfolge. Dabei ist $p = 1 - (1 - \alpha)^m$. Mit $p = 0,99$ und $\alpha = 0.001$ ergibt sich für $m \approx 4603$. D.h. es sind 4603 Folgen notwendig um daraus eine zu erhalten, die mit einer Wahrscheinlichkeit von 99% zu den Besten 0,1% gehört.

Mit diesem Vorgehen, das als *Ordinal Optimization* bezeichnet wird, lässt sich die Planungszeit deutlich reduzieren. Es ist offensichtlich, dass es keine Garantie gibt, die optimale Lösung zu erhalten. Aber wahrscheinlich wird eine gute gefunden. Die Grundidee, nicht alle möglichen Fußaufsetzpunkte zu evaluieren, soll in dieser Arbeit weiter verfolgt werden, um den Planungsaufwand zu reduzieren.

An der Arbeit von Chen und Kumar ist zu kritisieren, dass die Bewertung einer Roboterposition bzw. der Beinplatzierung für diese Position allein auf der Tragfähigkeit des Bodens basiert. Diese Fähigkeit wird durch Wahrscheinlichkeiten repräsentiert, deren Bestimmung nicht bekannt ist, was die Autoren auch anmerken. Da die physischen Fähigkeiten des Roboters bei der Bewertung keine Rolle spielen, erlaubt der Ansatz kein Übersteigen von Hindernissen.

5.4.2 Bai, Low und Teo (2002)

Eine Herangehensweise, um dieses Defizit beheben zu können, wird in Bai, Low und Teo (2002) vorgestellt. Grundlegend für diesen Ansatz ist die Verwendung einer 2½D-Karte der Umgebung, bestehend aus Gitterzellen. Entsprechend der Abbildung 5.4 wird ein Modell der Laufmaschine verwendet, welches die Reichweiten der Beine in so genannte *Arbeitsbereiche* einteilt. Ein Arbeitsbereich besteht aus einer rechteckigen Anordnung von Gitterzellen.

Bewertung des Geländes

Um eine Beurteilung der Gängigkeit des Geländes vornehmen zu können, werden die Hindernisse in zwei Klassen unterschieden: *Negative Hindernisse*, deren Höhe geringer ist als der mögliche Beinhub $L_{h,max}$, können vom Roboter überschritten werden (beispielsweise kleinere Steine, Löcher oder Gräben). *Positive Hindernisse* sind höher als $L_{h,max}$ und können nicht überschritten werden (z.B. Wände oder Klippen). Hindernisse werden dabei nicht direkt als solche erkannt, lediglich die Höhe(n) der entsprechenden Gitterzelle(n) fließen in die Bewertung ein. Kann ein Bein auf einer Gitterzelle platziert werden, so wird diese als *betretbar* bezeichnet. Andernfalls ist die Zelle *verboten*. Hierbei ist anzumerken, dass sowohl positive als auch negative Hinderniszellen als verboten betrachtet werden. Damit soll der „blockierende“ Effekt eines Hindernisses unterstrichen werden, da sich die Anzahl der möglichen Fußaufsetzpunkte reduziert. Allerdings

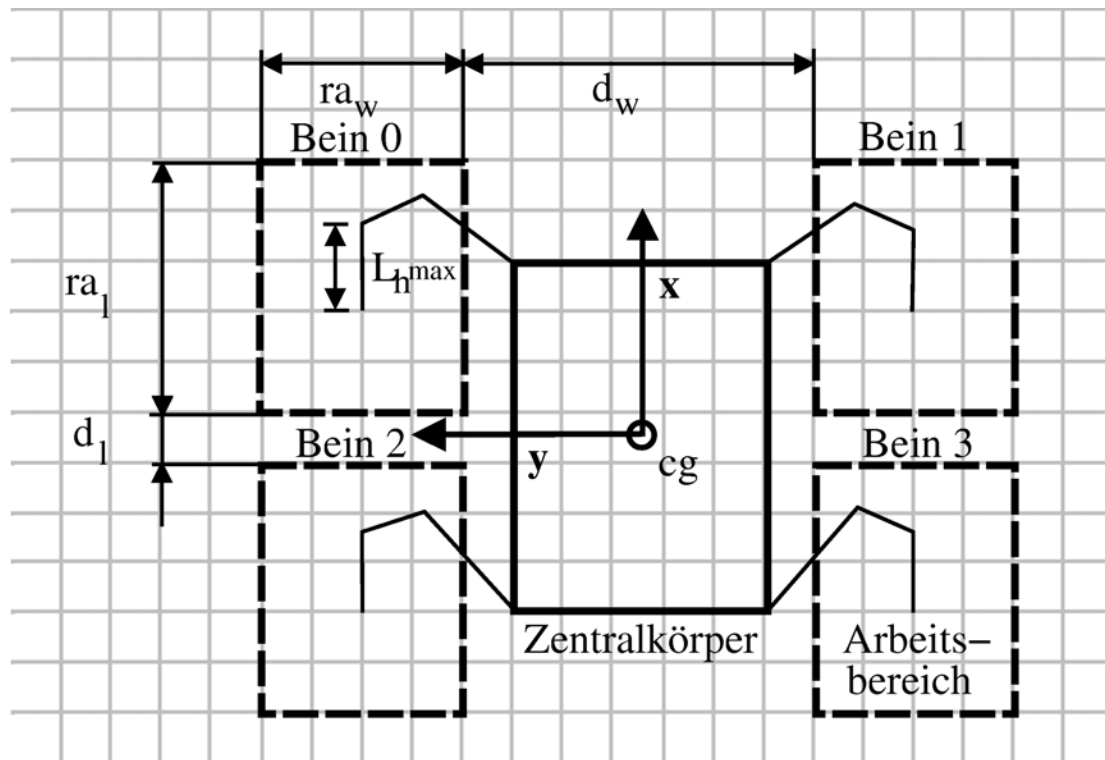


Abbildung 5.4: Laufmaschinenmodell: Zuteilung einer rechteckigen Anordnung von Gitterzellen zu jedem Bein. Der Beinhub $L_{h,\max}$ dient zur Klassifizierung von Hindernissen (nach Bai et al. (2002)).

besteht die Möglichkeit, dass einige der Nachbarn einer verbotenen Zelle betretbar sind. Aus dieser Option ergibt sich die so genannte **Zellenerreichbarkeit** A_c :

$$A_c(c) = \begin{cases} 1 & , \text{ wenn Zelle } c \text{ betretbar} \\ \frac{N_p}{N_t} & , \text{ sonst,} \end{cases} \quad (5.2)$$

wobei N_t die Anzahl der Zellen der Nachbarschaft ist, welche die Zelle c enthält. N_p ist die Anzahl der betretbaren Zellen unter den N_t Zellen.

Nun setzt sich der Arbeitsbereich eines Beines aus einer Menge von Zellen zusammen. Um für den Bereich eine Bewertung der Erreichbarkeit zu erhalten, überprüft man die Erreichbarkeit A_c jeder einzelnen Zelle. Es ergibt sich die **Bereichserreichbarkeit** A_a :

$$A_a = \frac{1}{N \cdot M} \sum_{i=1}^N \sum_{j=1}^M A_c(c), \quad c = c_{i,j}. \quad (5.3)$$

N und M geben dabei die Ausmaße eines Arbeitsbereiches in Zellen an. Ein Wert von $A_a = 1$ impliziert die Betretbarkeit aller Zellen. Währenddessen sind bei $A_a = 0$ alle Zellen verboten.

Die Laufmaschine bzw. deren Körpermittelpunkt cg erhält den Zugang zu einer bestimmten Gitterzelle, wenn ringsum ausreichend Fußaufsetzpunkte im Gelände für die Beine gefunden werden können. Es wird eine gemeinsame Aussage über die Bereichserreichbarkeiten aller n Beine benötigt, welche mit der **Geländezugänglichkeit** A_t

getroffen wird:

$$A_t(c) = \prod_{k=1}^n A_{a_k}, \quad c = c_{cg}, \quad (5.4)$$

wobei k der Index eines Beines ist. Diese Gleichung lässt sich als Wahrscheinlichkeit deuten, um eine bestimmte Zelle des Geländes erfolgreich betreten zu können. Des Weiteren ist sie ein Indiz für die Wegbeschaffenheit des Geländes. Ein hoher Wert für A_t spiegelt eine geringe Komplexität des Geländes wider. Um diesen Sachverhalt greif- und anwendbarer zu machen, wird daher die **Geländekomplexität** \overline{A}_t definiert:

$$\overline{A}_t(c) = 1 - A_t(c), \quad c = c_{cg}. \quad (5.5)$$

Somit setzt sich die Bewertung der Gitterzelle c_{cg} unterhalb des Körpermittelpunktes sukzessive aus den Bewertungen der umliegenden Zellen zusammen. Diese Herangehensweise ist allerdings nur bei negativen Hindernissen sinnvoll, da bei positiven die Erreichbarkeit benachbarter Zellen nicht garantiert werden kann. Um dem vorzubeugen, werden um positive Hindernisse so genannte *Einflussgrenzen* gelegt. Dabei wird jede Zelle des Hindernisses um r_m Zellen erweitert. Die Fusion aller erweiterten Zellen ergibt die Einflussgrenze. Befindet sich der Körpermittelpunkt innerhalb einer solchen Grenze, erfolgt automatisch eine Bewertung von $\overline{A}_t = 1$. Mit diesem Vorgehen gelingt den Autoren die Definition einer **3D-Geländekomplexität**:

$$\widetilde{A}_t(c) = \begin{cases} 1 & , \text{ wenn } c \text{ innerhalb einer Einflussgrenze} \\ \overline{A}_t(c) & , \text{ sonst.} \end{cases} \quad (5.6)$$

Ein hoher Wert ist gleichbedeutend mit einer komplexen und somit unwegigen Geländezelle. Der Wertebereich von \widetilde{A}_t ist dabei das Intervall $[0; 1]$.

Bahnplanung mittels Potenzialfeldern

Mit Hilfe der Gleichung 5.6 wird ein Potenzialfeld erzeugt. Dabei wird die 3D-Geländekomplexität zur Bildung eines abstoßenden Potentials verwendet:

$$U_r(c) = K_r \cdot \widetilde{A}_t(c), \quad (5.7)$$

während das anziehende Potenzial proportional zur Entfernung zum Ziel ist:

$$U_a(c) = K_a \frac{\|c - c_{goal}\|}{\|c_{start} - c_{goal}\|}. \quad (5.8)$$

Das Potenzialfeld setzt sich aus der Summe beider Einzelpotenziale zusammen. Die Konstanten K_r und K_a gewichten dabei das jeweilige Potenzial. Soll die Geländekomplexität eine größere Rolle bei der Planung spielen, so ist K_r größer als K_a zu wählen.

Die Planung erfolgt entlang des steilsten Gradientenabstiegs, d.h. ausgehend von der aktuellen Gitterzelle c wird immer die Nachbarzelle gewählt, deren Potenzial am geringsten ist. Um der Gefahr lokaler Minima vorzubeugen, bleibt der Roboter während des Planungsvorgangs solange an Ort und Stelle, bis ein Weg zum Ziel gefunden wird.

Während des Planungsvorgangs wird eine Baumrepräsentation aufgebaut, die alle bisher geplanten Wege beinhaltet. Somit kann ein Pfad, der in einem lokalen Minimum mündet, verlassen und ein alternativer Weg weitergeführt werden. Dies ist allerdings nur bei einer Offline-Planung möglich.

Beurteilung

Die Definition der Arbeitsbereiche für die Beine ist eine weit verbreitete Methode zur Bahnplanung für Laufmaschinen. Sie ermöglicht den Aufbau einer Bewertungsfunktion, welche die physikalischen Fähigkeiten des Roboters berücksichtigt. Allerdings wird nur der Hub einzelner Beine als Maß für die Unterteilung der Hindernisse in negative bzw. positive herangezogen. Grundlage dafür ist die absolute Höhe der Gitterzellen. Dies ist aber unzureichend, da die Wechselwirkung der einzelnen Beine untereinander nicht berücksichtigt wird. Daher ist auch die Definition der Einflussgrenzen und die Erweiterung auf 3D nicht zufrieden stellend. Der Hub der einzelnen Beine und die Höhe der Zellen müssen gemeinsam für die Beurteilung der Geländekomplexität herangezogen werden (Eldershaw und Yim, 2001). Nur so lässt sich z.B. eine Treppe als möglicher Weg erkennen.

Dennoch ist der sukzessive Aufbau einer Bewertung von einer elementaren Zelle bis hin zur Modellierung der 3D-Geländekomplexität sehr übersichtlich, effizient und erlaubt einen Austausch bzw. die Erweiterung einzelner Bewertungsschritte. Daher wird dieses Vorgehen als Grundgerüst für die vorliegende Arbeit dienen. Auch die Einflussgrenzen sollen übernommen werden, wenn auch mit veränderter Definition; erlauben sie doch die Einsparung von Komplexitätsberechnungen. Einhergehend mit dem Erhalt dieses Bewertungsschemas ist auch die Beibehaltung des Potenzialfeldes und dessen Planungsalgorithmus. Die Behandlung lokaler Minima ist eine sehr wünschenswerte Eigenschaft. Die Berücksichtigung unbekannter Umgebungen fehlt noch in diesem Ansatz und muss noch integriert werden. Es ist anzumerken, dass die Planung nur ein Orientierungszustand des Roboters berücksichtigt. Damit ist keine Rotation möglich, was die prinzipielle Unvollständigkeit des Verfahrens unterstreicht.

Offen bleibt die Verifikation des gefundenen Pfades. Die Autoren schlagen die Verfolgung mittels Simulation vor. Ein anderer Ansatz wird im Folgenden diskutiert.

5.4.3 Eldershaw und Yim (2001)

Ausgangspunkt ist, wie zuvor, eine $2\frac{1}{2}$ D-Repräsentation der Umgebung. Um die Verifikationsaufgabe bewältigen zu können, sieht der Ansatz von Eldershaw und Yim (2001) eine zweistufige Planung vor. Die High-Level-Planung bestimmt zunächst einen Weg, auf welchem der Roboter zum Ziel gelangen kann. Dieser Weg besteht aus einer Menge von Fußaufsetzpunkten, auf denen ein statisch-stabiler Stand des Roboters gewährleistet ist. Bei der nachgeschalteten Foot-Level-Planung wird eine Abfolge möglicher Beinbewegungen gesucht, damit der geplante Weg abgeschrieben werden kann. Gerade die zweite Stufe ermöglicht eine Aussage darüber, ob der Pfad tatsächlich gültig ist.

High-Level-Planung

Für diese Stufe wird, analog zum Ansatz im vorherigen Abschnitt, jedem Bein ein Arbeitsbereich zugeordnet. Zunächst erfolgt die Wahl eines Fußaufsetzpunktes für jedes Bein. Anschließend wird überprüft, ob diese Auswahl einen gültigen, stabilen Stand ergibt: Dieser ist dann gewährleistet, wenn der Zentralkörper durch die Höhe der Fußaufsetzpunkte und die Stemmhöhe der Beine nicht auf dem Untergrund aufsetzt. Wird der Stand als ungültig erkannt, bedarf es einer neuen Auswahl von Fußaufsetzpunkten, und zwar solange, bis geeignete oder keine Punkte gefunden werden. Im Falle einer gültigen Auswahl wird der Körpermittelpunkt als Teil des Weges gesichert, andernfalls als ungültig markiert. Mit diesem Vorgehen baut sich ein Weg sukzessive auf.

Foot-Level-Planung

Ob dieser Pfad allerdings von dem Roboter tatsächlich begangen werden kann, ist mittels der High-Level-Planung nicht zu entscheiden. Die Foot-Level-Planung untersucht, ob lediglich der Körper der Laufmaschine verschoben werden soll (die Beine bleiben stehen), oder ob ein Bein bewegt werden muss. Für den letzteren Fall sind zudem die Auswahl des Beines und der neue Aufsetzpunkt zu bestimmen. Wegen der mitunter großen Anzahl an Möglichkeiten steuern Heuristiken diese Suche. So wird z.B. das Bein bevorzugt, welches sich am Ende seines Arbeitsbereiches befindet und durch dessen Position der folgende Zustand ungültig werden würde. Da die Heuristiken den Suchraum zwar ordnen, aber nicht beschränken, ist die Planung auf dieser Stufe vollständig.

Beurteilung

Es wird ein vollständiger Ansatz vorgeschlagen. Er beinhaltet sowohl die Bahnplanung als auch deren Überprüfung durch die Foot-Level-Planung. Im Gegensatz zum Verfahren von Bai et al. (2002) erfolgt keine Unterteilung in überschreitbare und nicht überschreitbare Hindernisse. Mit der Berücksichtigung aller Aufsetzpunkte sowie der maximalen Beinhöhe kann eine Bewertung der Umgebung vorgenommen werden, ohne zuvor eine Hindernisklassifikation durchführen zu müssen. Da jedoch eine „Best-First“-Beurteilung des jeweiligen Körpermittelpunktes erfolgt, muss der resultierende Pfad nicht notwendigerweise der geeignetste sein. Hierfür bieten die Autoren Bai, Low und Teo durch ihren Potenzialfeldansatz eine Lösung. Beiden Verfahren gemein ist das Fehlen der Behandlung unbekannter Umgebungen.

Die Fußaufsetzpunktplanung zur Verifikation des gefundenen Wegs bietet zwar eine wünschenswerte Erweiterung, deren Berechnung erfordert aber einen hohen Zusatzaufwand. Wie schon in Abschnitt 5.1 erörtert, ist ein über diese Verifikationsaufgabe hinausgehender Nutzen auf Grund der erzielbaren Modellierungsgenauigkeit sehr zweifelhaft. Wählt man anstatt des erstbesten Wegs hingegen einen optimalen aus, sollte sich auch die Wahrscheinlichkeit eines real nicht begehbaren Wegs deutlich reduzieren. Nach Abwägen zwischen zusätzlichem Nutzen und notwendigem Rechenaufwand wird in dieser Arbeit auf eine Fußaufsetzpunktplanung verzichtet.

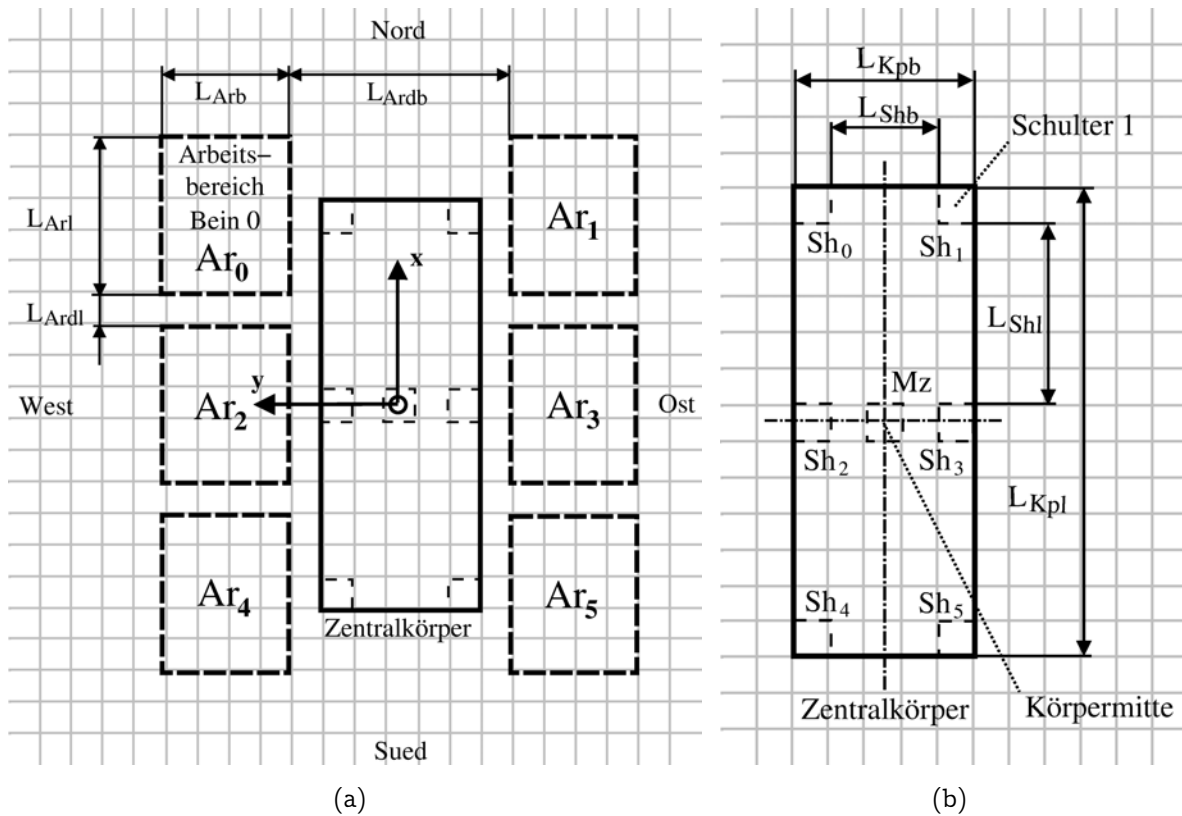


Abbildung 5.5: Zentralkörpermodell: Draufsicht mit Bearbeitsbereichen. (a) Die dick gestrichelten Rechtecke kennzeichnen die Bearbeitsbereiche Ar_i . Abgetragen sind deren Ausdehnung $L_{Ar1} \times L_{Arb}$ bzw. deren Abstände L_{Ardl} und L_{Ardb} zueinander, jeweils in Anzahl Zellen. Die x -Achse des Roboterkoordinatensystems zeigt in Richtung Nord, die y -Achse in Richtung West. (b) Die gestrichelten Zellen markieren die Körpermitte Mz und die Schultern Sh_i mit den Abständen L_{Shb} und L_{Shl} untereinander. Die strich-punkt-tierten Linien geben die Symmetrieachsen des L_{Kpl} langen und L_{Kpb} breiten Körpers an.

5.5 Generisches Laufmaschinenmodell

Die folgenden Abschnitte behandeln den im Rahmen dieser Arbeit entworfenen Bahnplaner unter Beachtung der Besonderheiten von Laufmaschinen.

5.5.1 Zentralkörpermodell

Zur Durchführung der Bahnplanung ist es notwendig, ein geeignetes Modell der Laufmaschine zu definieren. Die Abbildungen 5.5(a) und 5.5(b) skizzieren Draufsichten auf das Modell eines symmetrischen, L_n -beinigen Roboters mit identischen, insektenartig angeordneten Beinen. Der Fuß eines Beines i kann innerhalb des zugehörigen Arbeitsbereichs Ar_i beliebig platziert werden. Das Roboterkoordinatensystem liegt im Schnittpunkt der Symmetrieachsen, der Mittelzelle des Zentralkörpers Mz , mit diskretem Zellenindex (Mz_x, Mz_y) . Um den Planungsaufwand gering zu halten, werden die möglichen Orientierungen Mz_γ auf die vier Himmelsrichtungen Nord, West, Süd und Ost beschränkt.

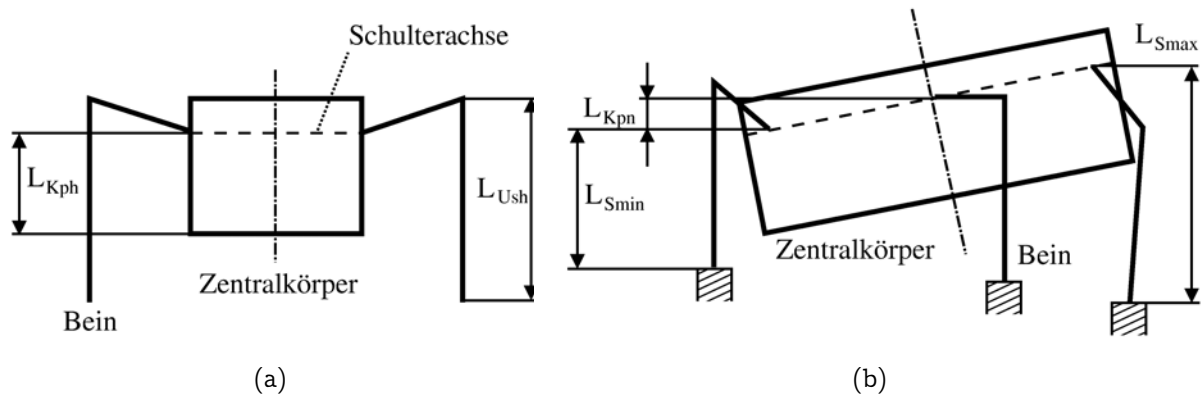


Abbildung 5.6: Zentralkörpermodell: Front- und Seitenansicht. (a) In dieser Frontansicht sind die Höhe des Zentralkörperunterteils L_{Kph} , die Lage der Schulterebene (gestrichelt) sowie die Unterschenkelhöhe L_{Ush} skizziert. (b) Diese Seitenansicht zeigt die Modellierung der Neigung L_{Kpn} um die y -Achse sowie die minimale L_{Smin} und maximale L_{Smax} Stemmhöhe der Beine.

Wegen der Symmetrie des Modells sind in späteren Überlegungen nur die beiden ersten Orientierungen zu berücksichtigen. Grundsätzlich lässt sich das Konzept leicht auf eine feinere Diskretisierung erweitern.

Es gilt die Annahme, dass sich alle Schultern Sh_i auf einer Ebene befinden. Diese wirkt als Nullebene und ist als gestrichelte Linie in Abbildung 5.6(a) eingezeichnet. Ab dieser wird die absolute Höhe L_{Kph} des unteren Teils des Zentralkörpers gemessen, womit das Aufsetzen des Körpers auf den Untergrund vermieden werden kann. Für das Beinmodell wird die Unterschenkelhöhe L_{Ush} eines Beins benötigt. Der Unterschenkel (lat. *Tibia*) ist kinematisch über ein Scharniergelenk mit dem Oberschenkel (lat. *Femur*) verbunden. Dieser mündet im Schultergelenk, welches drei Bewegungsfreiheitsgrade besitzt¹. Jedes Bein besitzt die gleiche minimale und maximale Stemmhöhe L_{Smin} bzw. L_{Smax} . Mit diesen Angaben lässt sich festlegen, wie groß der Abstand zwischen Fußaufsetzpunkt und Schulter minimal bzw. maximal durch ein Bein eingestellt werden kann. Bei einer negativen Angabe liegt die Schulter unterhalb des Fußaufsetzpunktes. Es gibt einen deutlichen Unterschied zwischen dem Beinhub L_{hmax} aus Bai et al. (2002) und den Stemmhöhen. Während der Beinhub voraussetzt, dass die Höhe des Körpermittelpunktes vorab bekannt ist, kann mit den Stemmhöhen die Höhe des Mittelpunktes beeinflusst werden. Das ist der wesentliche Grund, warum die 3D-Erweiterung bei Bai, Low und Teo nicht praktikabel ist.

Durch das Maß L_{Kpn} wird die maximale Rotation β um die y -Achse², die Neigungsfähigkeit des Roboters, angegeben. Es handelt sich dabei nicht um ein Winkelmaß, sondern um die Höhendifferenz zweier aufeinanderfolgender, so genannter *ipsilateraler* Schultern. Abbildung 5.6(b) zeigt die Abnahme des Neigungsmaßes. Die Fähigkeit einer Rotation α um die x -Achse³ wird nicht berücksichtigt. Dennoch wird eine Erweiterung zu Eldershaw und Yim (2001) erzielt, da dort keine Neigungsart vorgesehen ist.

¹Auf die Modellierung einer bei Insekten üblichen Hüfte (lat. *Coxa*) wird verzichtet.

²neigen oder im Englischen *pitch* genannt

³rollen oder im Englischen *roll* genannt

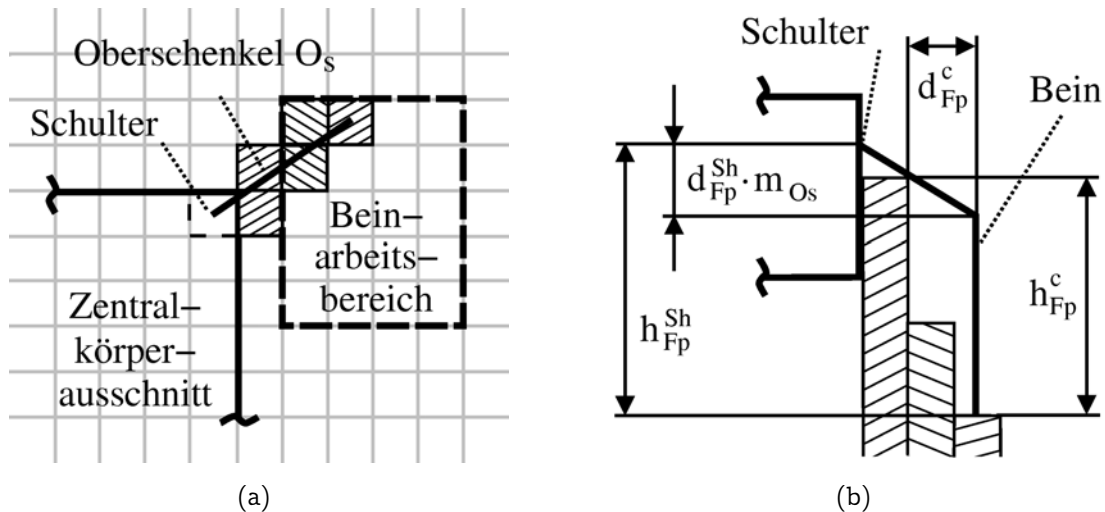


Abbildung 5.7: Beinmodell: Draufsicht und Frontansicht. (a) Die schraffierten Zellen in dieser Draufsicht werden von dem Oberschenkel O_s überquert. Diese Zellen kommen für das Aufliegen des Oberschenkels O_s in Frage. Der Unterschenkel U_s ist in der Projektion nicht sichtbar. (b) Der Oberschenkel kommt auf der linken Zelle c zum liegen. Deren seitlicher Abstand d_{Fp}^c zu und Höhe h_{Fp}^c über dem Fußaufsetzpunkt F_p legt die maximale Steigung m_{Os} fest. Mit dem Abstand d_{Fp}^{Sh} zwischen Schulter Sh und Fußaufsetzpunkt berechnet sich schließlich die minimale Höhe h_{Fp}^{Sh} der zugehörigen Schulter.

5.5.2 Beinmodell

In keinem der bisher beschriebenen Ansätze wird die Anatomie bzw. Kinematik des Beines genauer betrachtet. Lediglich Stemmhöhe oder Hub finden Berücksichtigung. Dabei treten nicht nur der Zentralkörper, sondern auch die Beine mit der Umwelt in Kontakt. Eine genauere Modellierung der Beine ist deswegen sinnvoll. Hierzu werden die folgenden Annahmen gemacht: Der Unterschenkel eines Beines stehe nahezu senkrecht und bleibe innerhalb der Umrise der Fußaufsetzpunktzelle. Durch entsprechende Wahl der Stemmhöhen und der Arbeitsbereichsgröße kann dies erreicht werden. Abbildung 5.7(a) zeigt die Positionierung des rechten, vorderen Beins in der Draufsicht. Der Oberschenkel überquert mehrere Zellen (schraffiert gekennzeichnet), welche potenziell die minimal mögliche Höhe der Schulterzelle begrenzen. In Abbildung 5.7(b) ist die *Auflagezelle* c des Oberschenkels dargestellt. Diese legt die maximale Steigung m_{Os} zum Knie fest, welche sich mit der Höhendifferenz h_{Fp}^c und dem seitlichen Abstand d_{Fp}^c zum Fußaufsetzpunkt F_p bestimmen lässt. Berücksichtigt man die Unterschenkelhöhe L_{Ush} , kann mittels der maximalen Oberschenkelsteigung und dem Abstand d_{Fp}^{Sh} zwischen Schulter und Fußaufsetzpunkt die vorläufige minimale Höhe der Schulter über dem Fußaufsetzpunkt mit

$$h_{Fp}^{Sh} = d_{Fp}^{Sh} \cdot m_{Os} + L_{Ush} \quad \text{und} \quad m_{Os} = \frac{h_{Fp}^c - L_{Ush}}{d_{Fp}^c} \quad (5.9)$$

angegeben werden. Darüber hinaus begrenzt auch die minimale Stemmhöhe $L_{S\min}$ das Absenken der Schulter, so dass für die relative minimale Schulterhöhe h_{Fp}^{Sh} und deren absolute Ausprägung h_a^{Sh} gilt:

$$h_{Fp}^{Sh} = \max(L_{S\min}, h_{Fp}^{Sh}) \quad h_a^{Sh} = h_{Fp}^{Sh} + h_a^{Fp}. \quad (5.10)$$

Durch die Neigungsmöglichkeit des Zentralkörpers ist zu beachten, dass sich die Lage der Schulterzelle verändern kann. Damit ergeben sich unterschiedliche Auflagezellen und somit auch veränderte Höhenwerte. Eine angebrachte Kombination der Höhen aller Lagen einer Schulter erlaubt eine Maximumbildung wie schon bei der Integration der minimalen Stemmhöhe in Gleichung 5.10. Für den weiteren Verlauf der Konzeption repräsentieren die Symbole h_{Fp}^{Sh} und h_a^{Sh} fortan auch deren Kombination.

Mit der Angabe aller aufgeführten Parameter ist das Modell der Laufmaschine vollständig beschrieben. Die in den folgenden Abschnitten erläuterten Konzepte bauen darauf auf.

5.6 Bewertungsfunktion

Das Herzstück jeder Bahnplanung ist die Grundlage, auf deren Basis entschieden wird, ob eine gewünschte Position, repräsentiert durch die Körpermittelpunktzelle und die Orientierung, für den Roboter gültig ist oder nicht. Dabei müssen zwar nicht alle Stellungen der Beine physikalisch möglich sein, einige aber schon. Da in Eldershaw und Yim (2001) durch die Vorgehensweise der High-Level-Planung ein Weg von Start zu Ziel auf einer Grundlage gebildet wird, die pro Gitterzelle des Weges mindestens eine gültige Stellung vorsieht, ist über die Qualität nichts ausgesagt. Im Sinne der Qualität von Roboterpositionen erfolgt die Planung, besonders anfangs, nicht zielgerichtet.

Daher sieht das Konzept der vorliegenden Arbeit in Anlehnung an Bai et al. (2002) ein Potenzialfeld vor, dessen Potenziale die Qualität einer Position repräsentieren. Dazu ist, neben dem bloßen Genehmigen oder Ablehnen einer Position, ein Zahlenwert vonnöten, der Auskunft darüber gibt, inwiefern und wieviel geeignete Stellungen gefunden werden konnten. Mit der Erhebung dieses Wertes, 3D-Geländekomplexität \tilde{A}_t genannt, beschäftigt sich dieser Abschnitt. Dabei sollen die in Abschnitt 5.4 besprochenen Schwächen des Verfahrens behoben, die Vorzüge jedoch erhalten bleiben. Die hierarchische Vorgehensweise ist äußerst praktikabel, die Erhebung der Potenzialwerte \tilde{A}_t bedarf hingegen einer Veränderung bezüglich der Entscheidung über die Gültigkeit der Stellungen der Beine. Die folgende Aufzählung gibt einen kurzen Überblick des hierarchischen Aufbaus der im Folgenden im Detail beschriebenen Bewertungsfunktion:

Stellungsmachbarkeit A_p : Beurteilung, ob eine Roboterlage bei fest gewählten Fußaufsetzpunktzellen gültig ist.

Geländezugänglichkeit A_t : Beurteilung, wie groß der Anteil der gültigen Stellungen unter allen möglichen Stellungen einer Roboterlage ist bzw. wie gut eine Stelle des Geländes für die Laufmaschine zugänglich ist.

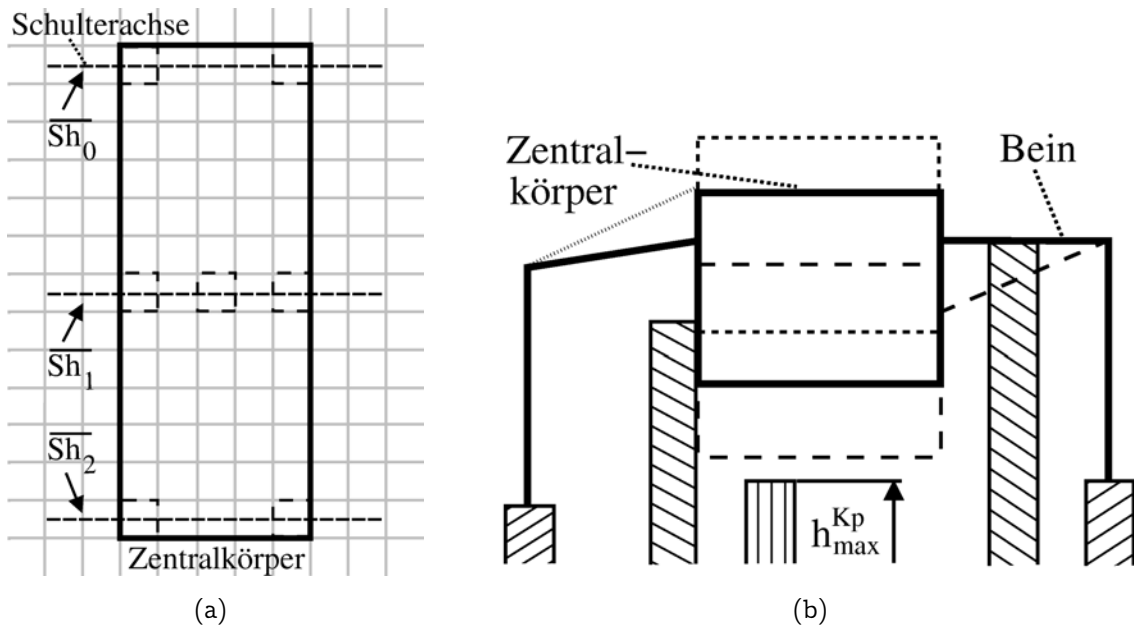


Abbildung 5.8: Minimale und maximale Höhe einer Schulterachse. (a) In der Draufsicht ist die Lage der Schulterachsen \overline{Sh}_j und deren Nummerierung durch gestrichelte Linien dargestellt. (b) Die gepunktete Körperstellung repräsentiert die maximale Höhe der Schulterachse. Durch das Hindernis bedingt, auf welchem das rechte Bein aufliegt, stellt die schwarze Lage die minimale dar. Ohne das rechte Beinhindernis ergäbe die gestrichelte Lage die minimale Höhe. Letztlich wird ein Absenken noch durch die höchste Zelle unterhalb des Körpers h_{\max}^{Kp} eingeschränkt.

Geländekomplexität \overline{A}_t : Negierte Interpretation der Geländezugänglichkeit.

3D-Geländekomplexität \widetilde{A}_t : Beschleunigung der Berechnung durch vorgeschaltete Analyse von Extremsituationen.

5.6.1 Stellungsmachbarkeit A_p

Hierfür sei angenommen, dass eine Roboterposition $c = (Mz_x, Mz_y, Mz_\gamma)$ und die Positionen der Beine gewählt seien. Alle Beine stehen auf dem Boden. Für einen statisch stabilen Stand reichen zwar schon drei Beine aus, dies wird jedoch im Sinne der einfacheren und schnelleren Berechnung außer Acht gelassen. Es wird eine generellere Formulierung der Stabilität zu Grunde gelegt (Eldershaw und Yim, 2001). Zudem sind für alle Beine i die Höhe $h_a^{\text{Fp}i}$ der Fußaufsetzpunktzelle und die minimale Schulterhöhe $h_a^{\text{Sh}i}$ bekannt.

Schulterachsen

Da eine Rotation des Zentralkörpers um seine x -Achse als nicht realisierbar angenommen wird, können benachbarte, gegenüberliegende Beine gemeinsam betrachtet werden. Sie bilden so genannte **Schulterachsen** \overline{Sh}_j aus (Abbildung 5.8(a)). Deren Index ergibt sich aus der Nummer des linken Beines dividiert durch 2.

Auf welchen Höhen sich *eine* Schulterachse bewegen kann, hängt von den Höhen der Fußaufsetzpunktzellen, den Schulterhöhen und den Stemmhöhen der beiden Beine ab. Da die niedrigere der beiden Fußaufsetzpunktzellen zusammen mit der maximalen Stemmhöhe $L_{S_{\max}}$ die maximale Höhe $h_{\max}^{\overline{Sh}_j}$ der Schulterachse j bestimmt, gilt:

$$h_{\max}^{\overline{Sh}_j} = \min_k (h_a^{Fp^k}) + L_{S_{\max}}, \quad k \in \{2 \cdot j, 2 \cdot j + 1\}. \quad (5.11)$$

Höher kann die Schulter j nicht gestemmt werden. Die Bestimmung der minimalen Höhe $h_{\min}^{\overline{Sh}_j}$ der Schulterachse j ist da ungleich schwieriger. Um die Vorgehensweise zu verdeutlichen, sind in Abbildung 5.8(b) die verschiedenen Möglichkeiten dargestellt. Sowohl Beinhindernisse als auch die minimale Stemmhöhe, beides in Gleichung 5.10 der minimalen Schulterhöhen $h_a^{Sh_i}$ berücksichtigt, können eine Ursache für das Minimum sein. Aber auch die Zelle unterhalb des Zentralkörpers mit maximaler Höhe h_{\max}^{Kp} beschränkt die Schulterachse nach unten. Berücksichtigt man die Höhe des Körpers bis zu den Schultern L_{Kph} , ergibt sich die minimale Höhe der Schulterachse j somit zu

$$h_{\min}^{\overline{Sh}_j} = \max(\max_k (h_a^{Sh_k}), h_{\max}^{Kp} + L_{Kph}), \quad k \in \{2 \cdot j, 2 \cdot j + 1\}. \quad (5.12)$$

Ist der Wert $h_{\min}^{\overline{Sh}_j} > h_{\max}^{\overline{Sh}_j}$, liegt entweder ein zu hohes Beinhindernis oder ein zu großer Höhenunterschied zwischen beiden Fußaufsetzpunkten vor. Die betrachtete Stellung kann physikalisch nicht realisiert werden. Weitere Berechnungen sind nicht erforderlich. Der Stellungsmachbarkeit A_p wird der Wert 0 zugeordnet.

Körperneigung

Gilt dagegen $h_{\min}^{\overline{Sh}_j} \leq h_{\max}^{\overline{Sh}_j}$, muss noch entschieden werden, ob der Zentralkörper die Höhen einnehmen kann, welche durch die Schulterachsen vorgegeben sind. Es ist also eine Gerade zu finden, die zwischen den minimalen und maximalen Höhen der Achsen liegt und dabei die maximale Neigung nicht überschreitet. Abbildung 5.9 skizziert das hierzu notwendige Vorgehen. Interessanterweise ist dabei die explizite Berechnung einer Geraden nicht erforderlich. Demnach ist:

$$A_p = \begin{cases} 1 & , \text{ wenn eine solche Gerade existiert} \\ 0 & , \text{ sonst.} \end{cases} \quad (5.13)$$

5.6.2 Geländezugänglichkeit A_t

Mit der Stellungsmachbarkeit A_p erhält man eine Aussage darüber, ob eine gewählte Konfiguration von Fußaufsetzpunkten eine gültige, d.h. physikalisch realisierbare Stellung ergibt. Nun kann an einer bestimmten Roboterposition c jeder Fuß in seinem Beinbereich beliebig platziert werden. Durch die Kombination aller Konfigurationen könnte ein Maß für die Wegbeschaffenheit des Geländes, die **Geländezugänglichkeit** $A_t(c)$, berechnet werden. Für die Normierung auf das Intervall $[0; 1]$ wäre die Summe der einzelnen Stellungsmachbarkeiten durch die Anzahl aller Kombinationen zu dividieren. Der Berechnungsaufwand hierzu ist erheblich: Bei 20 Zellen in einem Arbeitsbereich

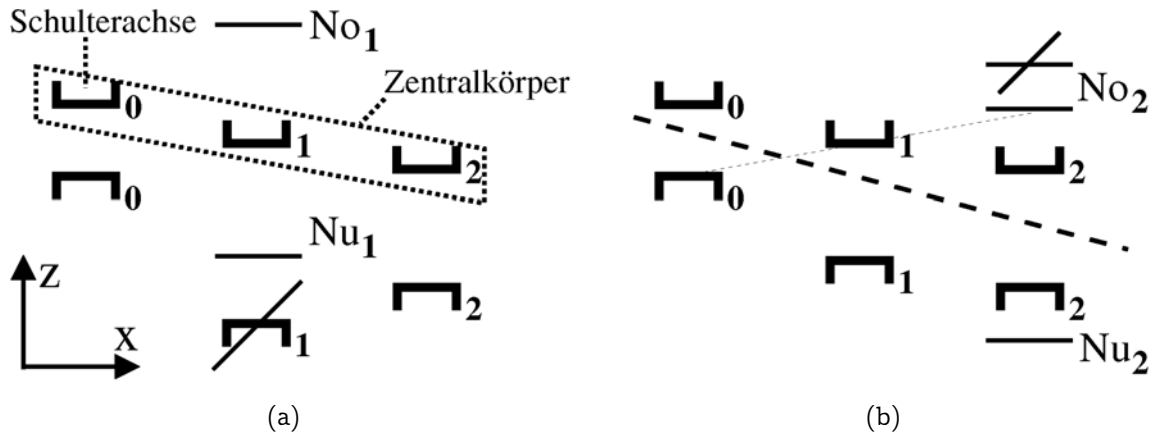


Abbildung 5.9: Die Schulterachsen realisieren eine gültige Stellung. (a) Dargestellt sind die minimalen \sqcup_j bzw. maximalen $\bar{\sqcup}_j$ Höhen der drei Schulterachsen \bar{Sh}_j . Wird zur minimalen (maximalen) Höhe von \bar{Sh}_0 die maximale Neigung L_{Kpn} subtrahiert (addiert), ergeben sich die feinen Striche bei \bar{Sh}_1 , die obere (untere) Neigungsschranke No_1 (Nu_1). Diese begrenzen die nachfolgenden Schulterachshöhen \sqcup_1 ($\bar{\sqcup}_1$). (b) Die aus $No_2 = \sqcup_1 + L_{Kpn}$ hervorgehende Neigungsschranke muss noch die Lage „zurückliegender“ Schulterachsen (hier Π_0) berücksichtigen, um eine Positionierung des Zentralkörpers garantieren zu können (gepunktete Linie). Schließlich findet sich mit der gestrichelten Geraden eine mögliche Zentralkörperstellung.

und 6 Beinen müssten 20^6 Stellungskonfigurationen untersucht werden. Und das bei einer einzigen Laufmaschinenposition! Zur Planung sind aber viele Roboterpositionen zu bewerten. Die Reduktion des Aufwands und den damit verbundenen Auswirkungen für die Geländezugänglichkeit wird im folgenden Abschnitt behandelt.

Unterteilung des Beinarbeitsbereichs in Zellflächen

Eine Möglichkeit, den Berechnungsaufwand zu reduzieren, ist die gemeinsame Überprüfung der Stellungsmachbarkeit von Zellen mit ähnlicher Höhe und ähnlicher minimaler Schulterhöhe. Zellen eines Beinarbeitsbereiches können zu einer Zellfläche \mathbb{ZF} zusammengefasst werden. Eine der Zellen $c_{\mathbb{ZF}} \in \mathbb{ZF}$, der so genannte *Repräsentant*, wird als Stellvertreter einer Zellfläche \mathbb{ZF} bestimmt. Seien c_k die Zellen eines Beinarbeitsbereiches mit den Höhen $h_a^{c_k}$ und den absoluten, minimalen Schulterhöhen $h_a^{Sh_{c_k}}$. Weiterhin sei $h_a^{\mathbb{ZF}}$ die Höhe des Repräsentanten und $h_a^{Sh_{\mathbb{ZF}}}$ seine Schulterhöhe, dann ist mit den gegebenen Toleranzen t_h und t_{Sh}

$$\mathbb{ZF} = \{c_k : |h_a^{c_k} - h_a^{\mathbb{ZF}}| \leq t_h \text{ und } |h_a^{Sh_{c_k}} - h_a^{Sh_{\mathbb{ZF}}}| \leq t_{Sh}\} \quad (5.14)$$

die Definition einer Zellfläche \mathbb{ZF} . Da die Höhen einer Zellfläche nicht mehr einheitlich sein müssen, können bei der Beurteilung der Stellungsmachbarkeit die Höhenwerte $h_a^{\mathbb{ZF}}$ und $h_a^{Sh_{\mathbb{ZF}}}$ des Repräsentanten herangezogen werden. Grundsätzlich ist diese Definition der Zellflächen von der Lage der Zellen innerhalb des Arbeitsbereichs unabhängig. Jedoch wird man in der Realisierung zusammenhängende Zellen favorisieren, da die Beurteilung einzelner Regionen dem Verständnis eines Fußaufsetzpunkts näher kommt.

Mit der Verwendung von Zellflächen anstelle einzelner Zellen kann die Anzahl an Erhebungen der Stellungsmachbarkeit A_p im Mittel deutlich gesenkt werden. Gerade bei großflächigen Ebenen lassen sich umfangreiche Zellflächen finden. Das Gelände kann allerdings auch sehr zerklüftet sein. Im schlechtesten Fall führt ein Beinarbeitsbereich mit 20 Zellen zu 20 Zellflächen, wenn alle Zellen unterschiedlich hoch sind. Hier kann keine Reduktion des Rechenaufwandes erreicht werden.

Kombination der Zellflächen

Damit sich auch für diese Fälle eine Reduktion erzielen lässt, bietet sich das Konzept der *Ordinal Optimization* an. Dessen Kernaussage ist, dass bei einer gegebenen Menge von Auswahlmöglichkeiten nicht notwendigerweise alle betrachtet werden müssen, um eine gute Konfiguration zu erhalten. Anstatt jede mögliche Kombination von Zellflächen verschiedener Beinarbeitsbereiche zu bestimmen, um für diese die Stellungsmachbarkeit zu überprüfen, reicht es möglicherweise aus, nur einige Kombinationen zu wählen.

Dafür wird die Menge M_i aller Zellflächen eines Arbeitsbereichs i zunächst in Teilmengen $T_{i,j}$ der Größe K zerlegt. K definiert dabei den *Kombinationsgrad*. Für jeden Arbeitsbereich i ergeben sich somit $n_i^T = \lceil \frac{|M_i|}{K} \rceil$ Teilmengen. Anschließend werden die folgenden Schritte durchgeführt, bis alle M_i leer sind:

1. Wähle für jeden Arbeitsbereich i eine Teilmenge $T_{i,j}$ aus.
2. Führe alle Kombinationen aus Zellflächen der gewählten Teilmengen aus. Bestimme für jede Kombination k die Stellungsmachbarkeit $A_p(k)$.
3. Entferne die gewählte Teilmenge $T_{i,j}$ jeweils aus *allen* Mengen M_i mit höchster Kardinalität n_i^T .

Abbildung 5.10 beschreibt diese Schritte an einem kleinen Beispiel.

Berechnung der Geländezugänglichkeit A_t

Entsprechend obigen Ablaufplans nimmt jede Teilmenge $T_{b,j}$ der Menge M_b mit maximaler Kardinalität $n_b^T = \max_i(n_i^T)$ genau an einem Kombinationsvorgang teil. Somit werden exakt n_b^T Kombinationsvorgänge durchgeführt. Die Anzahl an Kombinationen eines Kombinationsvorgangs j ist durch die Kardinalitäten der beitragenden Teilmengen bestimmt zu

$$m_j = \prod_{i=0}^{L_n-1} |T_{i,j}|, \quad \text{mit } |T_{i,j}| \leq K. \quad (5.15)$$

Da durch eine einzige Kombination $k_{j,l}$ ($l \in \{1, \dots, m_j\}$) indirekt auch alle Zellen der jeweils gewählten Zellflächen $ZF_i^{k_{j,l}}$ miteinander verglichen werden, repräsentiert eine einzige Kombination eine Vielzahl an Überprüfungen von Stellungsmachbarkeiten. Deren Anzahl berechnet sich zu

$$n_{k_{j,l}} = \prod_{i=0}^{L_n-1} |ZF_i^{k_{j,l}}|, \quad \text{wobei } ZF_i^{k_{j,l}} \in M_i. \quad (5.16)$$

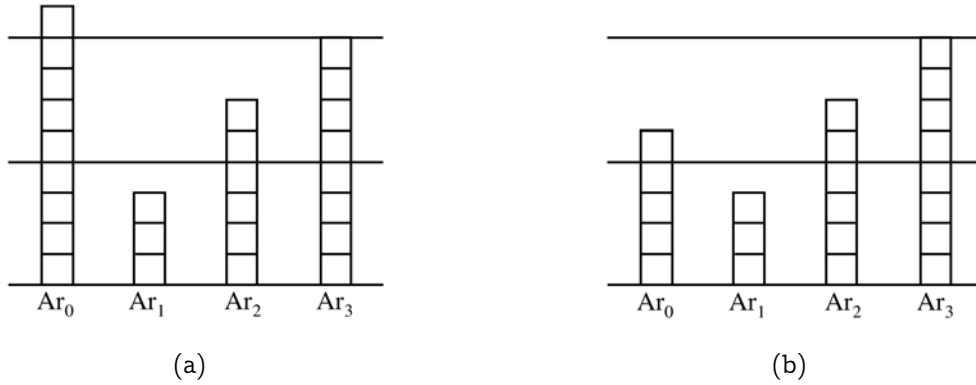


Abbildung 5.10: Ein Kombinationsvorgang mit vier Beinarbeitsbereichen Ar_i . (a) Jedes Kästchen repräsentiert eine Zellfläche ZF , jeder Stapel die Menge M_i aller Zellflächen. Der Kombinationsgrad ist mit $K = 4$ gewählt, was mit den horizontalen Linien angedeutet ist, welche die Teilmengen $T_{i,j}$ trennen. Es werden nur die untersten Teilmengen dem Kombinationsvorgang zugeführt. Alle damit möglichen Kombinationen $4 \cdot 3 \cdot 4 \cdot 4 = 768$ werden bestimmt. (b) Die Teilmenge $T_{0,1}$ des Arbeitsbereichs Ar_0 mit der höchsten Anzahl $n_0^T = 3$ an Teilmengen wurde nun entfernt. Es werden wieder die untersten Teilmengen $T_{i,2}$ miteinander vollständig kombiniert (hier gilt $T_{i,1} = T_{i,2} \forall i \in \{1, 2, 3\}$). Vor dem letzten Kombinationsvorgang sind die Arbeitsbereiche Ar_0, Ar_2 und Ar_3 zu reduzieren. Insgesamt führt dies zu $768 + 768 + 24 = 1560$ Kombinationen.

Mit diesem Wissen lässt sich die Geländezugänglichkeit einer Roboterposition c definieren:

$$A_t(c) = \frac{\sum_{j=1}^{n_b^T} \sum_{l=1}^{m_j} n_{k_{j,l}} \cdot A_p(k_{j,l})}{\sum_{j=1}^{n_b^T} \sum_{l=1}^{m_j} n_{k_{j,l}}} \quad (5.17)$$

5.6.3 Geländekomplexität \overline{A}_t

Mit dem Wert A_t aus Gleichung 5.17 erhält man eine Aussage darüber, wie gut eine Stelle des Geländes für die Laufmaschine zugänglich ist. Ein hoher Wert für A_t gibt an, dass für die Position c mehrere Stellungen der Beine realisierbar sind und daher das Gelände an dieser Stelle weniger *komplex* ist. Da auf der Grundlage von A_t eine abstoßendes Potenzial erzeugt werden soll, muss ein hoher Wert schlechtes Gelände charakterisieren. Diese Invertierung der Aussage gelingt durch die Definition der Geländekomplexität. Für die Position c errechnet sich die Geländekomplexität zu:

$$\overline{A}_t(c) = 1 - A_t(c). \quad (5.18)$$

Da der Wertebereich von A_t innerhalb des Intervalls $[0, 1]$ liegt, gilt gleiches auch für die Geländekomplexität \overline{A}_t . Mit dieser Funktion ist schon das Notwendigste vorhanden, um eine Potenzialfeldverfahren zu realisieren, wie es z.B. in Bai et al. (2002) formuliert worden ist. Allerdings kann diese Funktion erweitert werden, um in bestimmten Situationen eine schnellere Bewertung des Geländes und eine Selektion gewisser Hindernisse durchzuführen. Das Vorgehen beschreibt der folgende Abschnitt.

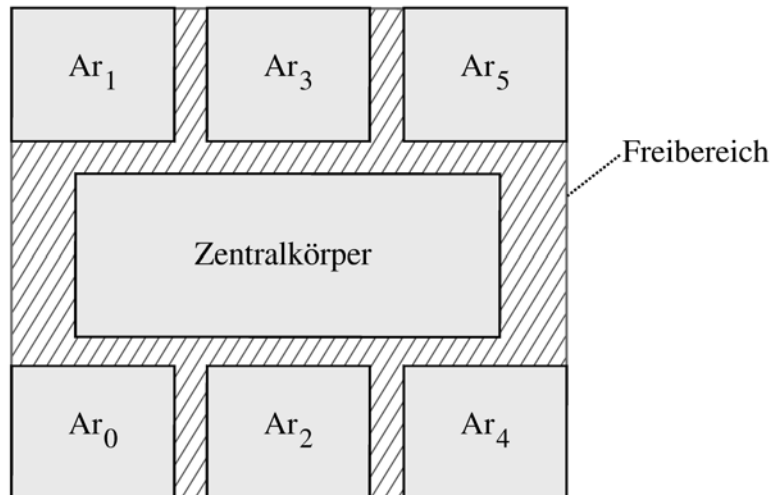


Abbildung 5.11: Das schraffierte Gebiet markiert den Freibereich \mathbb{F} der Laufmaschine.

5.6.4 3D-Geländekomplexität \widetilde{A}_t

In Abhängigkeit des Unterteilungsgrads K kann die Bewertung einer Roboterkonfiguration unterschiedlich viel Zeit in Anspruch nehmen. In unkritischem Gelände kann die Bildung von Zellflächen zu einer starken Reduktion der zu vergleichenden Fußstellungen führen. Ist das Gelände ungünstig, nimmt die Anzahl der Zellflächen zu, der Aufwand steigt. In manchen Situationen kann durch eine vorgeschaltete Analyse der aktuellen Konfiguration von dem skizzierten Bewertungsschema abgewichen und erheblicher Zeitaufwand eingespart werden.

Einflussgrenzen

Da für die Bahnplanung die Höheninformation zur Verfügung steht, kann diese genutzt werden, um Hindernisse großer Höhe sofort ihrer Unzugänglichkeit wegen zu meiden. Berücksichtigt man das Laufmaschinenmodell, dann wird von den Beinbereichen und dem Zentralkörper bzw. dessen Projektion auf den Untergrund eine Anzahl von Zellen überdeckt. Um eine Kollision des Roboters mit einem zu hohen Hindernis sicher vermeiden zu können, darf dieses Hindernis nicht innerhalb des Rechtecks gelangen, welches die Beinbereiche und den Körper umgibt. Das kleinst mögliche Rechteck umhüllt dabei die Laufmaschine gerade noch vollständig. Abbildung 5.11 zeigt dieses minimale umhüllende Rechteck. Bildet man die Schnittmenge aus den Zellen des Rechtecks mit den Zellen der Beinbereiche und des Zentralkörpers, erhält man den *Freibereich* \mathbb{F} .

Damit lässt sich die notwendige Analyse durchführen. In jedem Beinbereich Ar_i wird zunächst die höchste aller Zellen c bestimmt. Diese erhält das Symbol c_{\max}^i . Außerdem sei $c_{\max}^{\mathbb{F}}$ die höchste Zelle im Freibereich und c_{\max}^{Kp} die höchste Zelle unterhalb des Zentralkörpers. Die Höhen dieser Zellen sind entsprechend h_{\max}^i , $h_{\max}^{\mathbb{F}}$ und h_{\max}^{Kp} . Liegt nun eine der folgenden Situationen vor, befindet sich die Laufmaschine in einer *Einflussgrenze* eines zu hohen Hindernisses:

- $h_{\max}^{\mathbb{F}} > \max_i(h_{\max}^i) + L_{S\max}$ bzw. $h_{\max}^{\text{Kp}} > \max_i(h_{\max}^i) + L_{S\max}$
Selbst wenn alle Beine auf Zellen mit der Höhe $\max_i(h_{\max}^i)$ stehen würden und sich der Zentralkörper maximal anhebt, könnte das Hindernis im Freibereich bzw. unterhalb des Zentralkörpers nicht überschritten werden.
- $\max(h_{\max}^i, h_{\max}^j) + L_{S\min} > \min(h_{\max}^i, h_{\max}^j) + L_{S\max} + L_{\text{Kpn}} \cdot \left| \lfloor \frac{i}{2} \rfloor - \lfloor \frac{j}{2} \rfloor \right|$ mit $i \neq j$
Alle Beine befinden sich auf der höchsten Zelle in ihrem Arbeitsbereich: Der Höhenunterschied zwischen zwei der Beine kann selbst durch maximales Anheben des Zentralkörpers an einer Stelle durch das tiefer stehende Bein, maximales Absenken des Körpers an anderer Stelle durch das höher stehende Bein und Einstellen der maximalen Körperneigung nicht ausgeglichen werden.

Fallen alle Tests negativ aus, befindet sich die Laufmaschine nicht innerhalb einer Einflussgrenze eines zu hohen Hindernisses.

Komplexität von 3D-Gelände

Es liegt auf der Hand, dass durch diese Tests nur kritische Stellungen abgeprüft werden. Alle weiteren unterliegen der Überprüfung durch die Stellungsmachbarkeit. Damit ergibt sich für die Position c der Wert der 3D-Geländekomplexität zu

$$\widetilde{A}_t(c) = \begin{cases} 1 & , \text{ wenn } c \text{ innerhalb einer Einflussgrenze} \\ \overline{A}_t(c) & , \text{ sonst.} \end{cases} \quad (5.19)$$

Da im Fall einer Positionierung innerhalb einer Einflussgrenze lediglich die Überprüfung obiger Kriterien erfolgt, kann die Geländekomplexität sofort zu 1, dem schlechtesten Wert, bestimmt werden. Somit wird eine Beschleunigung der Bewertung erzielt. Kann kein Eindringen in eine Einflussgrenze festgestellt werden verläuft die Berechnung der Geländekomplexität nach dem skizzierten Schema. Der Mehraufwand der vorgeschalteten Einflussgrenzentests ist klein im Verhältnis zur Bestimmung des Wertes \overline{A}_t .

Damit ist die Bewertung der Zellen abgeschlossen. Sie resultiert in einem Wert aus dem Intervall $[0; 1]$, wobei 0 über eine Zelle (Mz_x, Mz_y) bei Orientierung Mz_γ aussagt, dass für sie jede Stellung machbar ist. Je höher die 3D-Geländekomplexität ausfällt, umso weniger Stellungen sind an der gegenwärtigen Position möglich. Mit dieser Bewertungsfunktion gelingt die Bildung eines abstoßenden Potentials, das sich zur Bahnplanung nutzen lässt.

5.7 Potenzialfeldverfahren

Analog zu Bai, Low und Teo (2002) setzt sich das resultierende Potenzialfeld aus der Überlagerung zweier durch die Faktoren K_r und K_a gewichteten Einzelpotenzialen zusammen: Dem direkt aus der 3D-Geländekomplexität gebildeten abstoßenden Potenzial

$U_r(c)$ und dem proportional zum Ziel hin ansteigenden Potenzial $U_a(c)$ (siehe Gleichungen 5.7 und 5.8):

$$U(c) = U_r(c) + U_a(c) = K_r \cdot \widetilde{A}_t(c) + K_a \frac{\|c - c_{\text{goal}}\|}{\|c_{\text{start}} - c_{\text{goal}}\|} \quad (5.20)$$

Der Gradientenabstieg im Potenzialfeld basiert auf dem in Bai et al. (2002) vorgestellten Algorithmus. Dieser realisiert eine *best-first*-Planung, bei der die erste Lösung bzw. der erste Weg zum Ziel als Planungsergebnis ausgegeben wird. Darüber hinaus ist die Konvergenz des Verfahrens sichergestellt: Es wird immer ein Weg zum Ziel gefunden, sofern ein solcher existiert. Eine maximal erlaubte Geländekomplexität $\widetilde{A}_{\text{max}}$ garantiert dabei die Begehbarkeit aller Zellen des gefundenen Pfades. Um die Effizienz und Planungsgenauigkeit zu steigern, werden nun noch einige Erweiterungen vorgeschlagen:

Zellbewertung nur auf Anforderung: Die Bewertung einer Zelle erfolgt nur dann, wenn sie auch tatsächlich beschriftet wird. Weiterhin wird die berechnete 3D-Geländekomplexität in der Datenstruktur der Zelle abgespeichert, um den Aufwand bei späteren Planungen zu reduzieren.

Mehrere Orientierungen: Für die betrachtete Laufmaschine werden vier diskrete Orientierungen berücksichtigt (Nord, West, Süd und Ost). Auf Grund der Symmetrie müssen nur zwei (Nord und West) zur Berechnung herangezogen werden. Das hat zur Folge, dass neben den vier Nachbarzellen, welche ausgehend von einer Zelle c durch Vorwärts- oder Seitwärtsbewegung mit der aktuellen Orientierung betretbar sind, noch die Zelle c selbst mit geänderter Orientierung zu bewerten ist. Dabei wird angenommen, dass sich der Roboter auf der Stelle drehen kann, sofern die Zellbewertung mit beiden Orientierungen akzeptabel ist.

Behandlung unbekannter Bereiche: Gelangt die Planung an einen Teil der $2\frac{1}{2}$ D-Karte mit wenig bekannten Zellen, ist darauf zu achten, dass die Bewertung nicht stark verfälscht wird. Vor der eigentlichen Bewertung wird deshalb untersucht, ob ausreichend viele bekannte Zellen im Einflussbereich der aktuellen Position der Laufmaschine vorhanden sind. In Analogie zur Bestimmung der Einflussgrenzen wird daher überprüft, ob in den Beinarbeitsbereichen ($R_{Ar} \geq \max_i(u_i)$), unterhalb des Zentralkörpers ($R_{Kp} \geq u_{Kp}$) sowie im Freibereich ($R_{F} \geq u_{F}$) der Anteil unbekannter Zellen (u_i , u_{Kp} bzw. u_{F}) unter einer vorgegebenen Schranke (R_{Ar} , R_{Kp} bzw. R_{F}) liegt. Schlägt eine dieser Überprüfungen fehl, befindet sich die Planung innerhalb oder in der Nähe eines unbekanntem Bereichs. Anstatt der Bewertung wird der Zelle ein zuvor festgelegter Explorationswert e zugewiesen. Wird dieser größer als A_{max} gewählt, kann die Zelle nicht Bestandteil eines Weges sein. Ist e kleiner, kann damit gesteuert werden, wie sehr unbekannte Bereiche bei der Wegsuche zu berücksichtigen sind. Mit $e = 0$ sind unbekannte und voll zugängliche Bereiche gleichwertig. Die Laufmaschine wird zur Erkundung der Umgebung „motiviert“.

5.8 Abschließender Vergleich

Mit den Erweiterungen des Potenzialfeldverfahrens sei die Konzeption des Bahnplanungsverfahrens für Laufmaschinen abgeschlossen. Die zum Aufbau eines abstoßenden Potenzials notwendigen Berechnungen erfolgen in einem mehrstufigen Verfahren: Zunächst wird die Verletzung von Einflussgrenzen untersucht. Ist dies nicht der Fall, kann die eigentliche Geländekomplexität \overline{A}_t bestimmt werden. Sie stellt dabei nur eine Invertierung der Geländezugänglichkeit A_t dar, um das abstoßende Element eines Potenzialfeldes aufzubauen. Die Geländezugänglichkeit gibt dabei das Verhältnis zwischen machbaren und allen möglichen Stellungen der Beine an. Da die Berechnung dieses Verhältnisses aufwändig sein kann, werden nicht einzelne Zellen als potenzielle Fußaufsetzpunkte betrachtet, sondern diese zu Zellflächen zusammengefasst. Weiterhin wird nicht jede mögliche Stellung ausgewählt. Mit dem Kombinationsgrad K lässt sich steuern, wieviele Kombinationen zwischen Zellflächen gebildet werden. Dadurch gibt die Geländezugänglichkeit den Anteil der gültigen unter allen betrachteten Stellungen wieder. Ob eine Stellung gültig ist, ermittelt die Stellungsmachbarkeit A_p . Sie liefert nur zwei Werte: Null, falls eine Kombination von Zellflächen möglich ist, andernfalls eins.

Tabelle 5.2 stellt den vorgeschlagenen Ansatz den Arbeiten von Bai, Low und Teo und Eldershaw und Yim gegenüber. Bei Bai, Low und Teo werden auf Grund des sehr einfachen Laufmaschinenmodells viele mögliche Wege nicht gefunden, was auch die Laufmaschinenanwendbarkeit dieses Ansatzes grundsätzlich in Frage stellt. Der vorgeschlagene Ansatz steigert die Anzahl der Wege bei gleicher Höhenkarte, lässt aber womöglich Wege zu, die der Laufmaschine nicht zugänglich sind. Wegen der Fußaufsetzpunktplanung ist der Ansatz nach Eldershaw und Yim dagegen gefeit, worunter jedoch die Rechereffizienz und Echtzeitfähigkeit zu leiden hat. Dadurch und wegen eines in manchen Teilen allgemeineren Modells ist dessen Laufmaschinenanwendbarkeit begründet, wenngleich weder Neigungsfähigkeit noch Beinhindernisse berücksichtigt werden.

Eine Korrekturmöglichkeit wird durch Bai, Low und Teo gar nicht behandelt. Die Fußaufsetzpunktplanung übernimmt bei Eldershaw und Yim das Erkennen irrtümlicher Wege, welche dann die High-Level-Planung korrigiert. Ein gefundener Weg ist auf Grund eines fehlenden Optimalitätskriterium jedoch nicht unbedingt der beste. Diese Arbeit behandelt eine einfache Konzeptionierung in Abschnitt 7.6.3.

Fehlende Begrenzungsflächen bereiten keinem der Ansätze Probleme. Unbekanntes Gelände spielt jedoch nur in der vorliegenden Arbeit eine Rolle.

5.9 Zusammenfassung

Die Aufgabe der Bahnplanung ist das Auffinden eines möglichst optimalen Wegs hin zu einem vorgegebenen Ziel. Auf dem Weg befindliche Hindernisse gilt es dabei zu vermeiden. Der Einsatz von Laufmaschinen ermöglicht jedoch das Überwinden von Hindernissen, wenngleich das Laufen in der Ebene zu bevorzugen ist. Der Bahnplaner sollte beides gebührend berücksichtigen, ohne den notwendigen Rechenaufwand aus den Augen zu verlieren. Unstrukturiertes Gelände erschwert die Planung zusätzlich: Große Teile der

	<i>vorliegende Arbeit</i>	<i>Bai, Low und Teo</i>	<i>Eldershaw und Yim</i>
Allgemein:			
Rechnereffizienz, Echtzeitfähigkeit	+	++	-
Vollständigkeit	+	o	++
Korrekturmöglichkeit	+	-	+
Unstrukturiertes Gelände:			
komplexe Hindernisse	++	-	+
keine Begrenzungsflächen	++	++	++
unbekanntes Terrain	++	--	--
Laufmaschinen:			
überwindbare Hindernisse	++	-	+
einfache Wege bevorzugen	++	++	-
Laufmaschinenanwendbarkeit	+	o	++

++: gut geeignet +: geeignet o: mäßig geeignet -: wenig geeignet --: ungeeignet

Tabelle 5.2: Gegenüberstellung verschiedener Ansätze mit Bezug auf die Anforderungen an Bahnplanungsverfahren für Laufmaschinen. Die vorliegende Arbeit wird als einzige allen Anforderungen gerecht.

Karte sind unbekannt; auch eine verbindliche Begrenzungsfläche kann nicht angegeben werden. Darüber hinaus muss mit komplexen Hindernisgeometrien gerechnet werden. Die grundlegenden Forderungen an Bahnplaner nach Korrekturmöglichkeiten und Vollständigkeit behalten auch in Bezug auf Laufmaschinen ihre Gültigkeit.

Diesen Anforderungen ist kaum ein Verfahren gewachsen: Der Einsatz von evolutionären Algorithmen verbietet sich auf Grund des enormen Aufwandes; andere Lernverfahren scheitern an unbekanntem Gelände. Auch auf Distance Transform basierende Verfahren sowie die Klasse der Dekompositionsverfahren können mit unbekanntem Gelände nicht umgehen; zusätzlich weisen sie Schwächen im Umgang mit komplexen, überwindbaren Hindernissen auf. Einzig die Potenzialfeldmethoden weisen genügend Flexibilität bei vertretbarem Rechenaufwand auf.

Der auf Potenzialfeldern basierende Ansatz von Bai et al. (2002) hat den in dieser Arbeit konzipierten Bahnplaner für Laufmaschinen stark inspiriert. Das von Bai, Low und Teo vorgestellte einfache Laufmaschinenmodell ist jedoch nicht ausreichend, um den vielfältigen Bewegungs- und Positionierungsmöglichkeiten einer Laufmaschine gerecht zu werden. Daher erfolgt die Erweiterung des Modells durch die Berücksichtigung der Körperneigung und der Stemmhöhen der Beine. Letztere ist durch Eldershaw und Yim (2001) motiviert und dient der Vermeidung einer zu simplen Klassifikation der Hindernisse auf Grund deren Höhe. Deren Ansatz vernachlässigt wiederum die Forderungen nach Optimalität und Rechereffizienz. Gerade im Hinblick auf die Verkürzung der Planungszeit greift diese Arbeit einige Ansatzpunkte auf. Trotzdem bleibt es weiterhin möglich, einen optimierten Weg durch unbekanntes, mit komplexen Hindernissen durchsetztes Gelände zu finden.

Kapitel 6

Adaption an das Gelände

Adaption (Adaptation) Anpassungsvermögen; Anpassung an die Gegebenheiten, Umstände, an die Umwelt.

(Fremdwörterduden, 1982)

Diese Definition aus dem Fremdwörterduden beschreibt knapp die an die Adaption gestellte Aufgabe: Die Laufmaschine soll sich selbstständig an die aktuellen Umweltbedingungen anpassen. Der Begriff der Adaption ist facettenreich: Von reflexartigen Reaktionen in Form direkter Sensor-Aktorkopplung (Cruse et al., 1998), über Anpassungen der Gangart (Weingarten et al., 2004) bis hin zum Wechsel der grundlegenden Fortbewegungsart (Asa et al., 2004) reicht das Spektrum alleine bei Laufmaschinen. Diese Arbeit fokussiert die Adaption – ganz im Gegensatz zur zielgerichteten Bahnplanung – im Sinne einer zeitnahen Planung: Die unmittelbar anstehenden Regelungsentscheidungen der Laufmaschine sollen durch Einsatz des im Umweltmodell gespeicherten Wissens unterstützt werden.

Im Folgenden werden Mechanismen präsentiert, die durch Interpretation der Modelldaten eine nachhaltige Anpassung des Laufverhaltens bewirken. Die in den Kapiteln zuvor genannten Anforderungen und Randbedingungen, insbesondere die Rechneffizienz betreffend, haben weiterhin Gültigkeit. Im Rahmen von Studien- und Diplomarbeiten (Bär, 2005) wurden hierzu wertvolle Beiträge geleistet.

6.1 Bewertung von Gelände

Die Grundlage zur Adaption des Laufverhaltens an die vorherrschenden Geländebedingungen wird durch eine Bewertung des Geländes geschaffen. Deshalb befasst sich dieser Abschnitt mit der Klassifikation der Geländebeschaffenheit und der Beurteilung der Begehbarkeit des Untergrundes.

6.1.1 Bewertung durch den Menschen

Eine Bewertung von Gelände steht immer in direktem Bezug zu den Lokomotions- bzw. Anpassungsfähigkeiten der Maschine, weshalb eine objektive Bewertung der Begehbarkeit von Gelände nicht möglich ist.

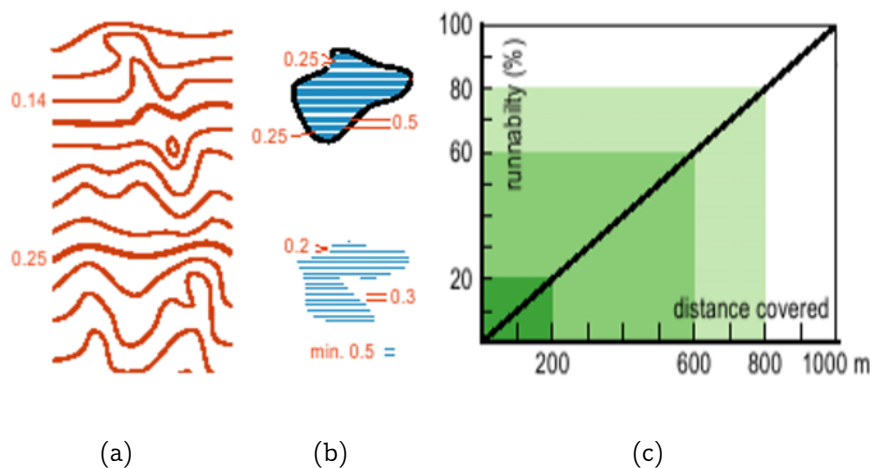


Abbildung 6.1: Orientierungslaufkarten. (a) In Orientierungslaufkarten sollte der Höhenunterschied zwischen zwei benachbarten Höhenlinien 5 Meter betragen. (b) Sumpfland wird gestrichelt dargestellt: je dünner schraffiert, desto besser passierbar. Ist es für den Läufer unpassierbar, wird es mit einer Linie umrandet. (c) Auf dunkleren Regionen kommt er mit 20% seiner Durchschnittsgeschwindigkeit voran; die hellen Regionen sind dagegen ohne Probleme zu begehen.

Geologen bewerten Böden vor allem nach Bearbeitbarkeit. Sie teilen dabei Boden und Fels nach DIN 18 300 in sieben verschiedene Klassen ein, wobei Klasse eins die am besten zu verarbeitende Bodenklasse (z.B. lockerer Waldboden) und Klasse sieben die am schwierigsten zu bearbeitende Klasse (z.B. Felsgestein) ist. Ein weiteres geologisches Kriterium ist die so genannte Korngrößenverteilung nach DIN 18 123. Zu deren Bestimmung werden aus einer Bodenprobe die Körner nach Größe getrennt und dann der prozentuale Anteil pro Korngröße anhand der Masse bestimmt. Nur selten werden Böden auf Begehbarkeit hin untersucht.

Die Verarbeitung geographischer Daten findet fast ausschließlich mit so genannten Geographischen Informationssystemen, oder kurz GIS, statt. Neben geometrischen Daten werden in einem GIS in der Regel weitere Daten gespeichert wie z.B. Straßen und Gebäude. Wegbeschaffenheitsinformation wie beispielsweise die Art des Straßenbelags kann in zusätzlichen Attributen verfügbar sein. Der für Laufmaschinen geforderte Detaillierungsgrad kann jedoch nicht erwartet werden.

Eine Bewertung des Untergrunds hinsichtlich der Begehbarkeit wird beim Orientierungslauf durchgeführt. Die Sportler müssen anhand einer Karte ein unbekanntes Gebiet in möglichst kurzer Zeit durchqueren. Das von der IOF Map Commission (2000) standardisierte Kartenformat legt dabei genau fest, welcher Maßstab zu wählen ist, wie Höhenlinien einzutragen sind, wie Wege zu kennzeichnen sind, ab welcher Breite ein Trampelpfad aufhört und ein Weg beginnt (Abbildung 6.1).

Leider gibt keines der aufgeführten Beispiele Aufschluss darüber, wie die Wegbeschaffenheitsinformation extrahiert werden kann. Sie wird in der Regel vom Menschen bewertet und in Karten eingetragen. Der nächste Abschnitt befasst sich deshalb mit Methoden zur automatisierten Bewertung von Geländeeigenschaften.

6.1.2 Maschinelle Bewertung

Wie schon in Kapitel 2 gezeigt, befassen sich nur wenige Arbeiten mit der Navigationsproblematik mobiler Roboter in natürlichem Gelände. In Bezug auf die Bewertung der Wegbeschaffenheit von Gelände verhält es sich ähnlich.

Bei größeren autonomen Fahrzeugen wie etwa Geländewagen stehen in erster Linie Angaben über den Verlauf des Weges und auftretende Hindernisse im Vordergrund (Coombs et al., 2000), (Patel et al., 2005). In natürlicher Umgebung werden auch Äste, Büsche, Bäume und „fester“ Untergrund sowohl auf Basis von geometrischen 3D-Daten als auch auf Basis von Farbkamerabildern klassifiziert (Vandapel et al., 2004), (Bellutta et al., 2000). Darüber hinausgehende Eigenschaften des Wegs oder eine feingranulare Klassifikation sind in der Regel nicht von Interesse.

Ein militärisches Aufklärungsfahrzeug ist hingegen speziell auf die Analyse von Geländecharakteristika ausgelegt – wobei nicht alle Auswertungen online durchgeführt werden (Shneier et al., 2004). Das mit Sensoren ausgestattete Geländefahrzeug wird durch unterschiedliche Gebiete geschickt, um die Befahrbarkeit des Geländes zu bewerten. Als Hauptdatenquelle des Bewertungsvorgangs dient ein 3D-Laserscanner. Shneier et al. ermitteln zuerst den festen Untergrund, um anschließend darauf stehende Bäume und Büsche zu detektieren. Schließlich wird versucht, die Oberflächensteigung und die Unebenheit festzustellen – wie die Autoren bemerken beides direkt von der gewählten Skalierung abhängig. Die Steigung wird aus der mit Hilfe der Kleinsten Fehlerquadrate gebildeten Ebene bestimmt, die Unebenheit durch die Standardabweichung der Messpunkte von dieser. Shneier et al. bemerken, dass diese Form der Bewertung auf zwei vom Menschen als unterschiedlich „schwierig“ zu befahren klassifizierten Routen statistisch vergleichbare Werte hervorbrachten. Eine andere genannte Methode basiert auf der Beobachtung der Fahrzeugneigung über die Zeit: Der Trend wird durch die Geländesteigung, die Schwankungen durch Unebenheiten hervorgerufen. Laufmaschinen können die Körperneigung ständig frei wählen.

Iagnemma et al. (2004) nutzen für die Geländeanalyse eines planetaren Rovers drei Komponenten: Die visuell ermittelte Einsinktiefe der Räder, die über die Rad-Boden-Interaktion berechnete Scherfestigkeit des Bodens sowie eine generelle Klassifikation des Bodens („sandig“, „steinig“, „gemischt“) anhand der Vibrationen an der Radaufhängung. Analog hierzu könnte auf Laufmaschinen die Einsinktiefe und das Rutschen der Beine zur Bewertung der Bodeneigenschaften herangezogen werden.

In Iagnemma und Dubowsky (2002) setzen die Autoren neben der Rad-Boden-Interaktion auch Audio- und Bildanalyse zur Geländeklassifikation eines schnell fahrenden Geländefahrzeuges ein (Abbildung 6.2). Bei langsamer Fortbewegung insbesondere auf Beinen scheidet die Audioanalyse jedoch aus. Eine bildbasierte Klassifikation des Geländes in grobe Klassen wie *fester Boden*, *lockerer Boden* oder *Vegetation* gibt nur begrenzt Aufschluss über die Begehbarkeit.

Letzteres vermeiden Seraji und Howard (2002) durch einen kontinuierlichen so genannten Wegbeschaffenheitsindex (engl. *traversability index*). Anhand von Stereo-Bildern wird unter Einsatz von Unschärfer Logik das Gelände nach den Faktoren Unebenheit (auf Basis der Größe und Konzentration von Steinen), Steigung sowie Unstetigkeit

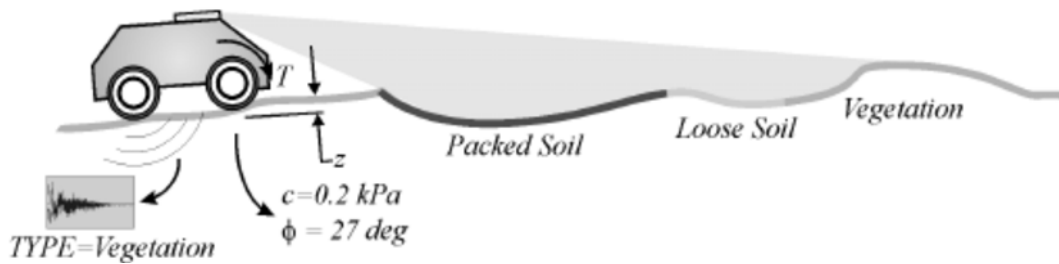


Abbildung 6.2: Iagnemma und Dubowsky (2002) kombinieren taktile Rad-Boden-Interaktion, Audioanalyse sowie Bildauswertung zur Geländeklassifikation eines schnell fahrenden Geländewagens.

bewertet. Die unscharfe Kombination dieser drei Qualitätsparameter bildet dann den Wegbeschaffenheitsindex, mit dessen Hilfe der Roboter schließlich navigiert. Aber auch dieser Ansatz zeigt, dass insbesondere eine kamerabasierte Beurteilung stark von Geländespezifika abhängig ist. Der Einsatz von Unscharfer Logik wird jedoch später noch einmal aufgegriffen werden, wenngleich nicht zur Bewertung von Gelände.

6.2 Schlussfolgerungen

Die Bewertung von Gelände hinsichtlich der Wegbeschaffenheit stellt sich als recht anspruchsvoller Themenkomplex heraus, der bei weitem noch nicht vollständig erforscht ist. Die wenigen Arbeiten, die sich mit dieser Problematik befassen, schränken sich auf das Nötigste ein. Selbst der Mensch tut sich mit einer objektiven Beurteilung von Gelände schwer.

Laufmaschinen können in der Regel weder Lageinformation noch Akustik zur Beurteilung heranziehen. Die Auswertung von Bildinformation mag sicherlich von Vorteil sein, kann man doch auf die vorherrschende Geländeart schließen. Eine geländeunabhängige Aussage über die Begehbarkeit der unmittelbaren Umgebung kann aber auch hier nur schwer getroffen werden.

Die Nutzung taktile Information ist auch bei Laufmaschinen denkbar. Die Beobachtung der Einsinktiefe und des Rutschens der Füße beim Auftreten bzw. während der Stemmphase könnten ein Indiz für die Qualität der Begehbarkeit sein. Auch Informationen direkt aus der Basissteuerung wie beispielsweise die Anzahl der aufgetretenen Beinkollisionen könnten hierzu beitragen: In Gaßmann (2000) wurde zu diesem Zwecke auf LAURON eine Bewertung der Untergrundbeschaffenheit durch Selbstreflexion der Beincoordination realisiert. Neben Beinkollisionen werden hierbei verfrühtes oder verzögertes Aufsetzen und reaktive Bodensuche berücksichtigt.

Die für LAURON zu realisierende Adaption stützt sich grundsätzlich auf zusätzlich aus dem geometrischen Modell zu ziehende Schlüsse. Weitergehende Untersuchungen hinsichtlich der Klassifikation von Gelände werden in dieser Arbeit nicht vorgesehen. Deshalb wird auf eine über die auf LAURON vorhandene Bewertung hinausgehende taktile Untergrundbeurteilung verzichtet. Im Folgenden steht die Beurteilung vornehmlich mit Hilfe geometrischer Geländeparameter im Mittelpunkt.

6.3 Beschreibung von Gelände

Dem letzten Abschnitt folgend, fällt die Wahl zur Geländebewertung auf den analytisch-geometrischen Ansatz. Ein Aufzeichnen aller Sensormessungen mit anschließender Nachbearbeitung, wie in Shneier et al. (2004) beschrieben, verbietet sich aus Effizienzgründen. Die Grundlage der Gebietsbewertung dieser Arbeit bildet die in Abschnitt 4.8 vorgestellte globale Höhenkarte. Dieser Abschnitt definiert die das Gelände beschreibenden Parameter, auf deren Basis später die Steuerungsentscheidungen gefällt werden.

Entgegen den meisten vorgestellten Verfahren zur Geländebewertung auf Basis von Laserdistanzdaten steht in dieser Arbeit kein hochauflösender 3D-Laserscanner mit hoher Reichweite zur Verfügung. Das verwendete 2½D-Modell der Umwelt wird Schritt für Schritt aufgebaut. Dies hat zur Folge, dass in der Karte nicht vermessene Gitterzellen existieren. Um diesem Umstand Rechnung zu tragen, wird zunächst die **Exploration** (*expl*) eines Gebietes \mathbb{G} definiert, als Anteil der in \mathbb{G} vermessenen Zellen c :

$$\text{expl}(\mathbb{G}) := \frac{|\mathbb{G}^+|}{|\mathbb{G}|} \quad , \quad \text{mit } \mathbb{G}^+ = \{c \in \mathbb{G} | c \text{ ist vermessen}\} \quad . \quad (6.1)$$

Des Weiteren steht aus der Sensordatenfusion Information über die Glaubwürdigkeit der gespeicherten Information zur Verfügung. Hiermit lässt sich die durchschnittliche **Zuverlässigkeit** (*cred*) eines Gebietes formulieren zu:

$$\text{cred}(\mathbb{G}) := \frac{1}{|\mathbb{G}^+|} \sum_{c \in \mathbb{G}^+} \text{cred}(c) \quad . \quad (6.2)$$

Mit Hilfe der Methode der Kleinsten Fehlerquadrate kann eine Ebene der Form $E : Ax + By + Cz = D$ in einem Gebiet bestimmt werden:

$$\sum_{c \in \mathbb{G}^+} (A \cdot p_x + B \cdot p_y + C \cdot p_z + D)^2 \longrightarrow \min \quad . \quad (6.3)$$

Anhand des Winkels der Kleinst-Quadrate-Ebene (kurz: KQ-Ebene) zur x - y -Ebene berechnet sich die durchschnittliche **Geländesteigung** (*slope*):

$$\text{slope}(\mathbb{G}) := |\angle(\mathbf{n}_E, \mathbf{n}_{xy})| \quad , \quad (6.4)$$

wobei $\mathbf{n}_E = (A, B, C)^T$ die Normale der KQ-Ebene E , $\mathbf{n}_{xy} = (0, 0, 1)^T$ die Normale der x - y -Ebene, also die z -Achse, und $\angle(\mathbf{a}, \mathbf{b})$ das orientierte Winkelmaß der Vektoren \mathbf{a} und \mathbf{b} darstellt.

Zur Beurteilung der *Unebenheit* des Gebiets \mathbb{G} werden abweichend von Shneier et al. (2004) neben der Standardabweichung der Punktabstände zur KQ-Ebene auch die statistischen Werte der Spannweite und des Medians dergleichen herangezogen. Die Spannweite gibt Aufschluss über die maximale, der Median über die mittlere auftretende Unstetigkeit. Die **Standardabweichung** (*dev*) des Gebiets berechnet sich zu:

$$\text{dev}(\mathbb{G}) := \sqrt{\frac{1}{|\mathbb{G}^+|} \sum_{c \in \mathbb{G}^+} d(c, E)^2} \quad , \quad (6.5)$$

mit $d(c, E)$ dem Abstand des Punktes c zur KQ-Ebene E . Schließlich ergeben sich die Spannweite (range) und der Median (med) aus:

$$\text{range}(\mathbb{G}) := \max_{c \in \mathbb{G}^+} (d(c, E)) \quad , \quad (6.6)$$

$$\text{med}(\mathbb{G}) := d_m(c, E) \quad , \quad \text{mit } m = \left\lfloor \frac{|\mathbb{G}^+| + 1}{2} \right\rfloor . \quad (6.7)$$

Hierbei steht $d_i(c, E)$ mit $i \in \{1, \dots, |\mathbb{G}^+|\}$ für den i .ten Index der geordneten Abstandswerte.

Hiermit schließt die mathematische Betrachtung des Geländes. Schließlich dient noch die taktile Untergrundbeurteilung der reaktiven Steuerung als weiterer Parameter zur Geländebeurteilung. Die im Folgenden vorgestellten Verfahren nutzen diese Beschreibung, um die Laufmaschine an die vorherrschenden Bedingungen anzupassen.

6.4 Auswahl von Laufparametern

Im letzten Abschnitt wurde ein Gebiet \mathbb{G} durch unterschiedliche Kennwerte beschrieben ($\text{expl}(\mathbb{G})$, $\text{cred}(\mathbb{G})$, $\text{slope}(\mathbb{G})$, $\text{dev}(\mathbb{G})$, $\text{range}(\mathbb{G})$, $\text{med}(\mathbb{G})$). Es bleibt nun zu klären, wie sich dieses Wissen nutzen lässt, um der Laufmaschine ein „gutes“ und „sicheres“ Durchqueren des Geländes zu ermöglichen. Das Laufverhalten des Roboters soll sich grundlegend an die vorherrschenden Geländebedingungen anpassen. Doch was bedeutet in diesem Zusammenhang „gut“ oder „sicher“? Und wie kann dies erreicht werden?

Bei Betrachtung des zu steuernden technischen Systems sind vor allem die folgenden Punkte von Bedeutung:

1. Welche Möglichkeiten zur Anpassung erlaubt eine Laufmaschine?
2. Auf welcher Basis soll eine Anpassung durchgeführt werden?
3. Und wie kann schließlich eine Steuerungsentscheidung herbeigeführt werden?

Eine Antwort auf die zweite Frage wurde schon im letzten Abschnitt gegeben. Mögliche Antworten auf die beiden anderen werden im Folgenden erörtert.

6.4.1 Laufparameter

Der mitunter wichtigste Parameter zur Steuerung der Fortbewegung, die Bewegungsrichtung, ist Gegenstand der Bahnplanung und steht nicht zur Disposition. Aber auch die Geschwindigkeit hat maßgeblichen Einfluss auf das Laufen: Bei geringer Geschwindigkeit bleibt dem Roboter mehr Zeit, auf Störungen wie beispielsweise unvorhergesehene Kollisionen zu reagieren oder durch Einsatz einer reaktiven Haltungskontrolle die Maschinenstabilität zu erhöhen (Gaßmann, 2000). Darüber hinaus erlaubt der Zeitgewinn, mehr Information über die nahegelegene Umwelt zu sammeln. Da die Laufmaschine sich jedoch möglichst schnell fortbewegen soll, muss zwischen „Schnelligkeit“ und „Sicherheit“ abgewägt werden. Die Geschwindigkeit von Laufmaschinen lässt sich gleich durch mehrere Parameter verändern:

Die **Schrittlänge** bzw. der Beinschlag (engl. stroke) bezeichnet die Strecke, die ein Bein während der Schwing- bzw. Stemmphase relativ zum Körper zurücklegt. Obgleich die „optimale“ Schrittlänge durch die Roboterkinematik gegeben ist, kann sie doch in Grenzen angepasst werden. Je größer die Schrittlänge, desto größere Strecken legt der Körpermittelpunkt innerhalb des verbleibenden Aufstandspolygons zurück. Der Abstand zu den Polygonrändern wird zeitweise geringer, so dass der Roboter grundsätzlich instabiler wird (vergleiche (Gaßmann, 2000)).

Die **Zykluszeit** ist die Zeitdauer, die ein Bein für einen kompletten Bewegungszyklus, also Schwing- und Stemmphase, benötigt. Je länger die Zykluszeit, desto langsamer und ruhiger werden die Roboterbewegungen.

Die **Gangart** bestimmt die grundsätzliche Art der Fortbewegung von Vier- und Mehrbeinern. Beispiele hierfür sind Schritt, Trab und Galopp beim Vierbeiner bzw. Tripod, Tetrapod und Pentapod beim Sechsheiner. Ein grundlegendes Merkmal der Gangart ist der so genannte *Belastungsfaktor* (engl. duty factor): Dieser sagt aus, welchen Anteil des Zyklus ein Bein in der Stemmphase verbringt. Beim Tripod, der Hauptgangart bei Insekten, beträgt der Belastungsfaktor $d = 0,5$; die Schwing- und Stemmphase der Beine sind gleich lang. Es wechseln sich immer 3 Beinpaare am Boden ab, daher auch der Name. Wird der Belastungsfaktor größer, erhöht sich die Anzahl der gleichzeitig am Boden befindlichen Beine. Bei $d = 1$ hebt kein Bein mehr vom Boden ab; der Roboter steht. Sinkt der Belastungsfaktor unter die Grenze von $d = 0,5$, beginnt der Roboter zu rennen – sofern er dazu überhaupt in der Lage ist.

Aus folgendem Zusammenhang zwischen Schrittlänge s , Zykluszeit t und Belastungsfaktor d lässt sich die Robotergeschwindigkeit v berechnen:

$$v = \frac{s}{d \cdot t} \quad . \quad (6.8)$$

In schwierigem oder unbekanntem Gelände sollte der Roboter seine Geschwindigkeit verlangsamen, also die Schrittlänge reduzieren, die Zykluszeit verlängern sowie den Belastungsfaktor erhöhen. Auf ebenem Grund kann der Roboter hingegen schneller voranschreiten und entsprechend größere Schritte bei kleinem Belastungsfaktor und kurzer Zykluszeit wählen.

Neben der Geschwindigkeit lassen sich noch weitere Laufparameter mit Einfluss auf das Laufverhalten identifizieren:

Die **Schritthöhe** legt fest, wie weit das Bein in der Schwingphase vom Boden abhebt. In rauem Gelände ist es angebracht, diese zu erhöhen, um möglichen „Stolperfallen“ schon im Vorfeld zu entgehen. Darüber hinaus wird beim Wiederabsetzen des Fußes ein größerer vertikaler Bereich bei steiler Bahnkurve abgedeckt. Dies unterstützt das Finden des Untergrunds vor allem bei Geländeunebenheiten.

Die **Schrittbreite** bestimmt den seitlichen Abstand der Füße zum Zentralkörper. Obwohl die Schrittbreite in der Regel durch die Beinkinematik festgelegt ist, kann

diese doch oftmals in engen Grenzen variiert werden. So bietet ein breiterer Stand eine erhöhte Aufstandsfläche und somit zusätzliche Stabilität. Dies kann unter schwierigen Rahmenbedingungen von Vorteil sein.

Die Körperhöhe legt den Abstand des Zentralkörpers vom Untergrund bzw. von den Fußaufsetzpunkten fest. Auch diese ist nicht bei jedem Laufroboter beliebig variierbar. Jedoch steigert ein Absenken des Schwerpunktes im Allgemeinen die Stabilität der Maschine; insbesondere Steigungen können mit niedrigem Körper besser gemeistert werden.

Die Körperposition definiert im Gegensatz zu Körperhöhe die Verschiebung des Zentralkörpers in der Horizontalen. Ergibt eine derartige Anpassung in der Ebene nur beim Rennen Sinn, so erhöht eine Verschiebung des Zentralkörpers in Richtung einer Steigung die Roboterstabilität doch ungemein.

Die Körperneigung gibt die Orientierung des Zentralkörpers vor. Bei leichten Unebenheiten und geringen Steigungen ist es dem Roboter möglich, sich waagrecht bzgl. der Gravitation auszurichten. Je größer jedoch die Steigung, desto stärker werden die Bewegungsfreiräume der vorderen und hinteren Beine eingeschränkt; der Aufstieg wird beschwerlicher. Die allmähliche Rotation in die parallel zur Steigung verlaufenden Ebene kann hier Abhilfe schaffen. Jedoch reduziert dies wiederum die Stabilität des Roboters, da die Projektion des Schwerpunktes hangabwärts wandert.

Es sei angemerkt, dass viele Robotersteuerungen eine Haltungskontrolle auf reaktiven Steuerungsebenen vorsehen, welche die Position, die Höhe und vor allem die Neigung des Zentralkörpers selbstständig regelt. Aber auch hier kann die Vorgabe von Stellgrößen Berücksichtigung finden – sofern dadurch die Stabilität der Maschine nicht gefährdet wird.

In diesem Sinne verhält es sich ebenfalls mit den anderen hier vorgestellten Laufparametern: Die Adaption spricht auf Basis des gesammelten Umweltwissens eine Art Empfehlung aus, welche durch die darunterliegende Laufmaschinensteuerung entweder umgesetzt oder auf Grund höher priorisierter Interessen ignoriert werden kann.

6.4.2 Unscharfe Logik zur Parameterwahl

Welches Laufverhalten in welcher Umgebung unter welchen Bedingungen das optimale ist, lässt sich oft nicht eindeutig feststellen. Diese Arbeit beschränkt sich daher auf das oben angedeutete einfache Vorgehen: In schwierigem Gelände soll der Roboter langsam und vorsichtig sein; einfaches Gelände kann er hingegen zügig durchqueren.

Wie man diese umgangssprachliche Formulierung einer expliziten Regelungsentscheidung zuführen kann, wurde schon an anderer Stelle aufgezeigt¹: Mit Methoden der

¹Sowohl die Beingewichtung bei der Odometrie (Abschnitt 3.4.2) als auch die Bewertung der Gitterzellen des 3D-Umweltmodells (Abschnitt 4.7.4) greifen auf Methoden der Unscharfen Logik zurück. Anhang B gibt hierzu eine kurze Einführung.

Unschärfe Logik lassen sich Regeln aufstellen, die linguistische Terme wie „steil“, „wenig“ oder „groß“ linguistischer Variablen wie „Steigung“ oder „Schritthöhe“ miteinander verknüpfen.

Als Eingangsgrößen dienen die in Abschnitt 6.3 hergeleiteten Kennwerte: Exploration, Zuverlässigkeit, Geländesteigung sowie Standardabweichung, Spannweite und Median der Bodenunebenheit. Darüber hinaus steht die in Abschnitt 6.2 angesprochene Bodenbewertung durch Selbstreflexion der reaktiven Laufmaschinensteuerung zur Verfügung. Die zu steuernden Laufparameter aus dem vorigen Abschnitt bilden die Ausgangsgrößen: Zykluszeit, Belastungsfaktor, Schrittlänge, -höhe, -breite, Körperhöhe, -position sowie -neigung. Als Regeln kommen beispielsweise folgende zum Einsatz:

$$\begin{array}{ll}
 \text{WENN } \text{expl}(\mathbb{G}) = \text{sehr klein} & \text{DANN } \text{Zykluszeit} = \text{hoch} \\
 \text{WENN } \text{slope}(\mathbb{G}) = \text{groß} & \text{DANN } \text{Belastungsfaktor} = \text{hoch} \\
 \text{WENN } \text{dev}(\mathbb{G}) = \text{mittel} & \text{DANN } \text{Schrittgröße} = \text{mittel} \\
 \text{WENN } \text{med}(\mathbb{G}) = \text{groß} & \text{DANN } \text{Schritthöhe} = \text{groß}
 \end{array} \quad (6.9)$$

Anhang C.3 gibt einen Überblick bezüglich des kompletten Regelsatzes. Durch die Nutzung von Fuzzy-Logik kann die Bodenbewertung durch Selbstreflexion (Einsinken, Rutschen, Anzahl aufgetretener Beinkollisionen) leicht in den Fusionsprozess integriert werden.

Das vorgestellte Verfahren passt das Laufverhalten des Roboters nachhaltig an die Umgebungsbedingungen an. Der nächste Abschnitt betrachtet Adaption im Sinne der Platzierung des nächsten Schritts.

6.5 Dediziertes Setzen von Fußpunkten

Verzichtet die Bahnplanung auf Grund bestehender Modellierungsunsicherheiten auf eine Fußaufsetzpunktplanung zur Verifikation des geplanten Weges, so kann die lokale Planung auf verlässliche Erkenntnisse zurückgreifen: Die unmittelbare Umgebung vor dem Roboter wird fortlaufend vermessen und der aktuellen Roboterlage entsprechend in die Karte eingetragen. Bis Ende des nächsten Schritts halten sich die auftretenden Ungenauigkeiten in Grenzen.

Somit besteht die Möglichkeit, auf Basis des gesammelten Umweltwissens den folgenden Fußaufsetzpunkt geeignet zu selektieren. Auf Grundlage des vom Bahnplaner ermittelten Weges und den aktuell gewählten Laufparametern (vergleiche Abschnitt 6.4) berechnet der Roboter, wohin der Fuß beim nächsten Schritt in etwa bewegt werden würde. Anschließend wird ein Ausschnitt des Weltmodells rund um dieses Zielgebiet betrachtet: Ist nicht genügend verlässliche Information zur Hand, kann durch gezieltes Nachmessen zunächst die Zuverlässigkeit dieses Kartenbereichs erhöht werden (vergleiche Abschnitt 7.6.3). Ansonsten wird, ausgehend vom geplanten Fußaufsetzpunkt, ein geeigneter Ort zur Platzierung des Fußes gesucht (Abbildung 6.3). Ein *guter* Fußaufsetzpunkt muss dabei

1. begehbar sein,

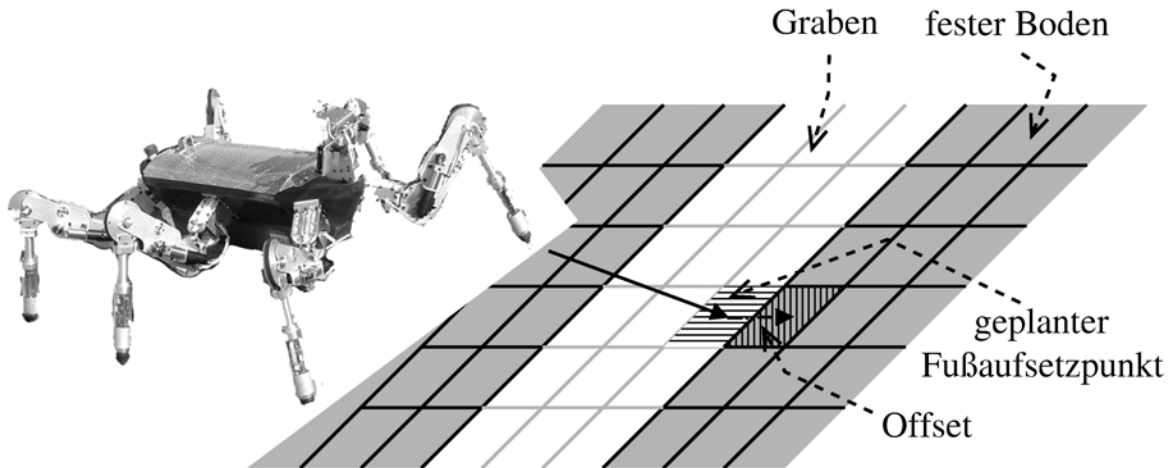


Abbildung 6.3: Finden eines geeigneten Fußaufsetzpunktes: Der voraussichtliche neue Fußaufsetzpunkt für das linke Vorderbein (horizontal schraffiert) befindet sich in einem Graben (weiß). Aus der Umweltinformation erweist sich die im Bild rechts daneben liegende Zelle als geeignetere Aufsetzstelle.

2. nahe an dem voraussichtlichen Fußaufsetzpunkt liegen,
3. und mit hinreichender Zuverlässigkeit bekannt sein.

Eine Zelle $c = (x, y)$ gilt als nicht begehbar, wenn sie in der $2\frac{1}{2}D$ -Ausschnitts-Karte nicht definiert ist, oder der Höhenunterschied zur aktuellen Fußposition eine Schwelle Δh_{\max} überschreitet. Nichtbegehbaren Zellen sei im Weiteren ein Zuverlässigkeitswert von $\text{cred}(c) = 0$ zugeschrieben. Zur Bewertung des *optimalen* Aufsetzpunktes c_{opt} muss man noch zwischen Nähe zum voraussichtlichen Fußaufsetzpunkt c_{plan} und der Zuverlässigkeit abwägen:

$$c_{\text{opt}} = \arg \max_c (\text{cred}(c) \cdot (1 - K \cdot \|c_{\text{plan}} - c\|)) \quad . \quad (6.10)$$

Der Faktor K bestimmt dabei, wie stark Vorgabennähe und große Zuverlässigkeit gegeneinander gewichtet werden sollen. Ein sinnvoller Wert wäre beispielsweise

$$K = \left(\frac{\|c_{\text{plan}} - c\|}{\max_c \|c_{\text{plan}} - c\|} + 1 \right)^{-1} \quad . \quad (6.11)$$

Diese Wahl bewirkt, dass Punkte an den Rändern der Karte eine doppelt so hohe Zuverlässigkeit aufweisen müssen wie der Mittelpunkt, um berücksichtigt zu werden. Mit der Bewertungsfunktion für den optimalen Aufsetzpunkt schließt die Konzeption der Adaptionenmechanismen.

6.6 Zusammenfassung

Eine Adaption von Laufmaschinen an die Bedingungen in ihrer nahen Umgebung kann auf viele Arten durchgeführt werden. Anliegen dieser Arbeit ist es, zu demonstrieren,

welche über die reine Bahnplanung hinausgehenden Möglichkeiten sich durch den Einsatz eines Umweltmodells bieten.

Erkenntnisse über die nahe gelegene Umwelt sind zur Entscheidungsfindung unabdingbar. Das umliegende Gelände muss hierfür geeignet bewertet und klassifiziert werden. Neben geometrischen Daten, wie sie schließlich in dieser Arbeit Verwendung finden, kommen in der Literatur vor allem bildverarbeitende Verfahren zum Einsatz, die im Allgemeinen auf spezifische Problemfelder spezialisiert sind. Es wurde erörtert, wie andere innovative Methoden zur Bewertung der Wegbeschaffenheit zukünftig auf Laufmaschinen übertragbar sein könnten.

Auf Grundlage ausgewählter Kenngrößen des vorliegenden geometrischen 2½D-Modells werden das Laufverhalten beeinflussende Steuerungsparameter bestimmt. Intuitive Regelsätze der Unscharfen Logik werden aufgestellt, um das Laufen auf das aktuelle Gelände abzustimmen. Letztlich wird die unmittelbar bevorstehende Aufgabe einer Laufmaschine, das Setzen des nächsten Schrittes, durch das im Umweltmodell gespeicherte Wissen unterstützt.

Es sei angemerkt, dass viele darüber hinausgehende vorausschauende Planungsmethoden denkbar sind. Zukünftige Arbeiten seien beispielsweise motiviert, sich der Planung einer optimalen Schrittfolge zur Überwindung von extremen Hindernissen zu stellen.

Kapitel 7

Realisierung, Experimente und Ergebnisse

In den vorangehenden Kapitel wurde eine modellbasierte, sensorgestützte Navigation für Laufmaschinen im Gelände konzipiert. Dieses Kapitel befasst sich nun mit deren konkreten Realisierung auf der autonomen Laufmaschine LAURON IV.

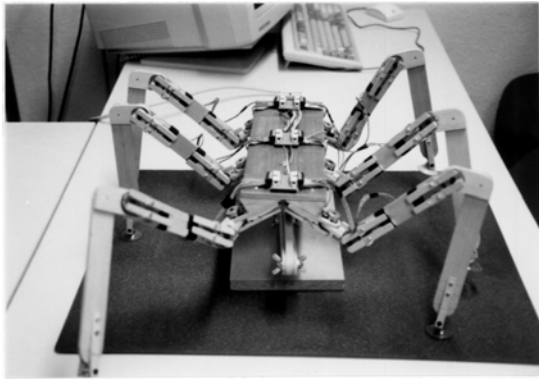
Hierzu wird zunächst die sechsbeinige Laufmaschine LAURON IV mit ihrem mechanischen Aufbau, der zur Verfügung stehenden Rechnerhardware sowie der vorhandenen Sensorik vorgestellt. Besondere Aufmerksamkeit wird den navigationsrelevanten Sensoren zuteil. Anschließend wird die reaktive Steuerung von LAURON umrissen. Den darauf aufsetzenden Navigationsalgorithmen und deren Einbettung in das Gesamtsystem sind die weiteren Abschnitte gewidmet.

An geeigneter Stelle wird die Funktionstüchtigkeit des realisierten Navigationssystems und dessen Komponenten anhand durchgeführter Experimente verifiziert und deren Ergebnisse diskutiert. Hierbei soll gezeigt werden, dass die entwickelten Algorithmen das Gewünschte leisten und deren spezielle Ausprägungen zu einer echten Verbesserung des Systemverhaltens führen. Um der Komplexität des Gesamtsystems gerecht zu werden, werden die unterschiedlichen Teilaspekte schrittweise einer detaillierten Analyse unterzogen, um schließlich die Fähigkeiten des Gesamtsystems unter Beweis zu stellen.

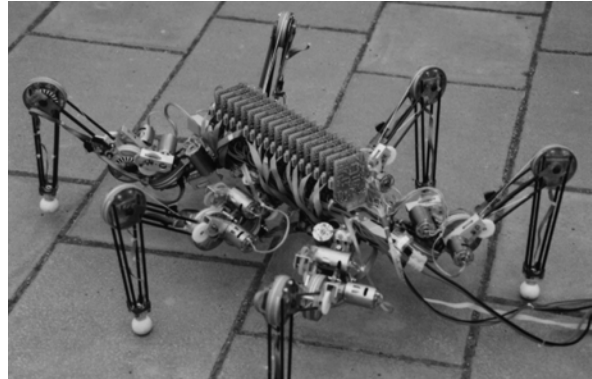
7.1 Die sechsbeinige Laufmaschine Lauron IV

Seit Anfang der 90er Jahre wird am *Forschungszentrum Informatik (FZI)* sechsbeiniges Laufen untersucht. Nach ersten Untersuchungen in der Simulation und am Holzmodell INSECT (Abbildung 7.1(a)) wurde 1994 die sechsbeinige Laufmaschine LAURON I entwickelt (Abbildung 7.1(b)). Der Name LAURON leitet sich aus der damaligen Steuerungsmethode ab: LAUFender Roboter, Neuronal gesteuert.

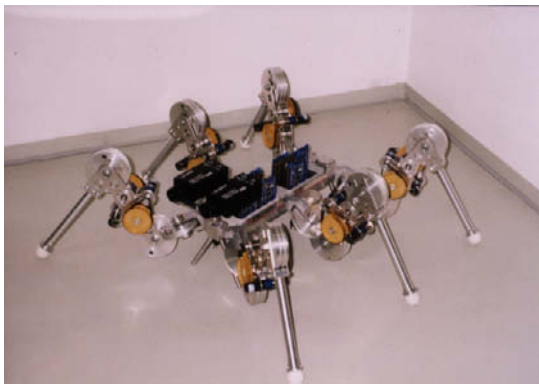
In zweiter Generation erhielt LAURON II 1996 nach kompletter Überarbeitung der Mechanik eine neue Rechnerarchitektur sowie verbesserte Sensorik (Abbildung 7.1(c)). Weitere mechanische Optimierungen und Erweiterungen wie beispielsweise eine 3D-Kraftsensorik an den Füßen wurden in der dritten Generation 1999 realisiert (Abbildung 7.1(d)). Alle vorgeschlagenen Steuerungskonzepte wurden im Rahmen von Studien- und Diplomarbeiten auf LAURON III umgesetzt (Gaßmann und Berns, 2002; Fromm-



(a)



(b)



(c)



(d)



(e)



(f)

Abbildung 7.1: Die LAURON-Ahnengalerie. (a) INSECT: Holzmodell zur Erzeugung und Aufzeichnung von Trajektorien (Cordes, 1992) (b) LAURON I: Testplattform für maschinelle Lernverfahren zur Bewältigung von Steuerungsproblemen und zur Sensorauswertung (Cordes, 1994) (c) LAURON II: Robustere Mechanik durch Einsatz von Aluminium (Cordes und Bührlé, 1996) (d) LAURON III: Erweiterte Sensorik erlaubt verbesserte Umwelterkennung (Kepplin und Berns, 1999) (e) LAURON IV: Technische Zeichnung (f) LAURON IV: Erprobung im Freien

berger, 2002; Gaßmann et al., 2003a,b; Wiedemann, 2003; Gaßmann et al., 2004; Zacharias, 2004; Gaßmann et al., 2005b; Huber, 2005; Gaßmann et al., 2005a; Bär, 2005; Haug und Rühl, 2005; Gaßmann et al., 2006). Im September 2005 wurde schließlich LAURON IV fertiggestellt (Abbildungen 7.1(e) und 7.1(f)) und diente als Demonstrator zur Evaluierung der vorgeschlagenen und entwickelten Methoden.

7.1.1 Technischer Aufbau

LAURON IV besteht aus einem Zentralkörper aus Kohlefaser an dem sechs gleichartige Beine symmetrisch befestigt sind. Jedes Bein besteht aus drei Segmenten, die jeweils durch ein Gelenk miteinander verbunden sind. Mit je drei Bewegungsfreiheitsgraden pro Bein ergeben sich insgesamt 18 aktiv zu steuernde Bewegungsfreiheitsgrade, mit Hilfe derer sich der Roboter fortbewegen kann (Abbildung 7.1(e)).

Im normalen Stand hat LAURON IV die Ausmaße von $90 \times 100 \times 40 \text{ cm}^1$ und wiegt mit eingebauten Akkumulatoren 20 kg. Die integrierte Energieversorgung reicht für einen autonomen einstündigen Betrieb. Am vorderen Ende des Zentralkörpers ist ein Sensorkopf mit zwei Bewegungsfreiheitsgraden angebracht, um Sensorik wie beispielsweise Kameras oder Laserscanner auszurichten.

Die Basisregelung der Beine und des Kopfes übernimmt jeweils ein so genanntes Universal Controller Module (UCoM), eine Rechneinheit aus DSP und FPGA, in Verbindung mit einer Aufsteckkarte zur Motoransteuerung (Regenstein und Dillmann, 2003). Die sieben UCoM-Einheiten sind über CAN-Bus mit einem robusten PC/104-Kompaktrechner verbunden. Dessen 933 MHz Pentium-III Prozessor definiert den Leistungsrahmen für die zu realisierende modellbasierte Navigation auf LAURON IV. Mit dem Betriebssystem Linux und dessen echtzeitfähiger Erweiterung RTAI² ausgestattet muss der Zentralrechner allen Steuerungsanforderungen genügen.

Zur Erfassung der internen Zustände stehen LAURON IV folgende Sensoren zur Verfügung:

Gelenkwinkel: optische Encoder mit Auflösung von 4000 Impulsen pro Umdrehung; mit Hilfe eines Index-Kanals kann beim Systemstart initial die absolute Gelenkstellung ermittelt werden

Gelenkgeschwindigkeit: Motorencoder mit 16 Impulsen pro Motorumdrehung

Motorstrommessung: Direkt auf den UCoM-Einheiten wird die von den Motoren verbrauchte Leistung gemessen.

Rundumkamera, Stereokamerasystem: Auf dem Roboterrücken ist eine Rundumkamera, im Roboterkopf ein Stereokamerasystem untergebracht. Diese unterstützen den Benutzer bei der Fernsteuerung (Albiez et al., 2005) (Abbildungen 7.2(b) und 7.2(d)).

Die weitere Sensorik der Laufmaschine steht im engeren Bezug zu dieser Arbeit, weshalb diese in den folgenden Abschnitten gesondert vorgestellt wird.

¹Die Aufbauten auf dem Rücken von LAURON haben eine zusätzliche Höhe von etwa 40 cm.

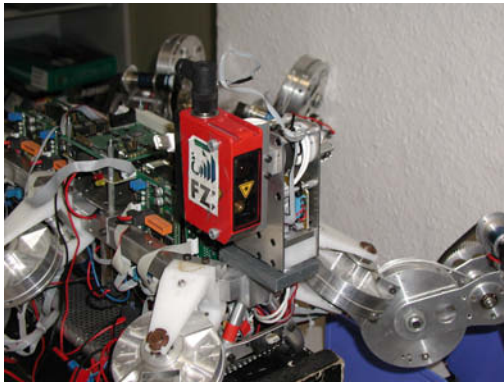
²Real-Time Application Interface for Linux <http://www.rtai.org>



(a)



(b)



(c)



(d)

Abbildung 7.2: Sensorik zur Umwelterfassung von LAURON IV. (a) Der Fußkraftsensor mit Auswertungs elektronik. (b) Der erhöht angebrachte Orientierungssensor (schwarze Box) und die Rundumkamera an einer Halterung mit aufgesetzter GPS-Antenne. (c) Der über weite Teile der Arbeit eingesetzte 1D-Abstandslaser auf LAURON III. (d) Der aktuelle 2D-Abstandslaser sowie das Stereokamerasystem auf dem Roboterkopf.

7.1.2 Fußkraftsensoren

Die Odometrie dient als ständig verfügbares, relativ messendes Lokalisationsverfahren. Dabei sieht die Konzeption der Odometrie von Laufmaschinen wie LAURON eine Gewichtung der Beine vor (siehe Abschnitt 3.4). Neben den Gelenkwinkelgebern zur Berechnung der Fußpunkte im Roboterkoordinatensystem (direkte Kinematik) sind hierbei vor allem die Fußkraftsensoren von zentraler Bedeutung: Zu welchem Zeitpunkt findet der relevante Übergang von Schwing- zu Stemmphase wirklich statt? Und zu welchem Teil sind die Stemmbeine zu gewichten?

Den Anforderungen dieser Arbeit folgend, wurde LAURON IV im Vergleich zu LAURON III mit leistungsfähigen Kraftsensoren ausgestattet. An definierten Biegestellen werden Dehnmessstreifen in Halb- bzw. Vollbrücke beschaltet (x -, y - bzw. z -Komponente), um die den auftretenden Kräften proportionalen Verformungen zu messen. Die Auswertelektronik zur Verstärkung, AD-Wandlung, Kalibrierung, Skalierung und Filterung befindet sich direkt im Unterschenkel: über ein serielles Kommunikationsprotokoll werden die Daten an den jeweiligen UCoM weitergeleitet (Abbildung 7.2(a)).

Zur Steuerung von LAURON IV bedarf es Methoden der Fuzzy-Logik, um die geforderten feinen Abstufungen zu erreichen. Hierzu wurde ein mehrstufiges Verfahren aus Bodenkontakt-, Kollisions- und Rutsch-Bewertung auf der unteren Ebene sowie einer Gesamt-Bewertung zur Gewichtung konzipiert (Wiedemann, 2003). Es konnte aber gezeigt werden, dass eine fundierte Bodenkontaktbewertung wesentlich bessere Ergebnisse ermöglicht (Haug und Rühl, 2005).

Für die Auswertung der Fußkraftsensoren mittels Unscharfer Logik sind im ersten Schritt die einfließenden linguistischen Variablen mit den zugehörigen Fuzzy-Mengen zu spezifizieren. Die Konzeption in Abschnitt 3.4.2 beschreibt die am Fuß wirkende Kraft als eine der wichtigsten Informationsquellen. Experimente haben gezeigt, dass die in Richtung der Gravitation wirkende Kraftkomponente Fz_j^a eines Beins j eine gute Wahl zur Bewertung des Bodenkontaktes darstellt. Die Berücksichtigung der anderen Kraftkomponenten verfälscht das Ergebnis, da beispielsweise bei Kollisionen der Füße oder bei Verspannungen die horizontal wirkenden Kräfte ansteigen, jedoch ein sicherer Kontakt zum Boden gar nicht besteht oder dieser verringert wird. Abbildung 7.3 erläutert die erzielten Ergebnisse anhand eines repräsentativen Datenverlaufs über der Zeit.

Da eine sechsbeinige Laufmaschine unterschiedlich viele Beine gleichzeitig am Boden halten kann, fällt die Definition aussagekräftiger Fuzzy-Mengen für die linguistische Variable *z-Kraft* schwer, obgleich ab einer gewissen Schwelle auf sicheres Auftreten geschlossen werden kann. Um bei kleineren Kräften feinere Abstufungen zu erlauben, wird deshalb der relative *z-Kraftanteil* an der *z-Gesamtkraft*, der *z-Kraft-Quotient* Fz_j^r , eingeführt:

$$Fz_j^r = \frac{Fz_j^a}{\sum_j Fz_j^a} \quad . \quad (7.1)$$

Anhang C.1 listet die letztlich verwendeten Fuzzy-Mengen und -Regeln auf.

Neben der Nutzung als taktile Sensor für die reaktive Steuerung und die Odometrie-berechnung lassen Beine mit ausreichender Bodenkontaktbewertung auch den Schluss zu, dass sich unter ihnen fester Boden befindet. Eine weitere Schlussfolgerung besagt, dass am Platz des Fußes selbst kein belegter Boden zu finden ist. Somit kann diese Information auch in das Umweltmodell eingetragen werden.

7.1.3 Orientierungssensor

Das Lokalisationskonzept in Kapitel 3 sieht die Verwendung einer absoluten Referenz vor, um den durch die Odometrie wachsenden Orientierungsfehler zu beschränken. Deshalb wurde LAURON mit einem Orientierungssensor ausgestattet, dem Microstrain 3DM-G³. Es handelt sich hierbei um eine Kombination von drei Drehratengyroskopen, drei orthogonalen Beschleunigungsmessern, drei orthogonalen Magnetfeldsensoren und einem eingebetteten Mikrocontroller. Ein Temperatursensor zum Messen der geräteinternen Temperatur ist ebenfalls vorhanden. Die Daten können jeweils in unveränderter oder in so genannter gyroskopisch-stabilisierter Form angefordert werden.

³<http://www.microstrain.com/>

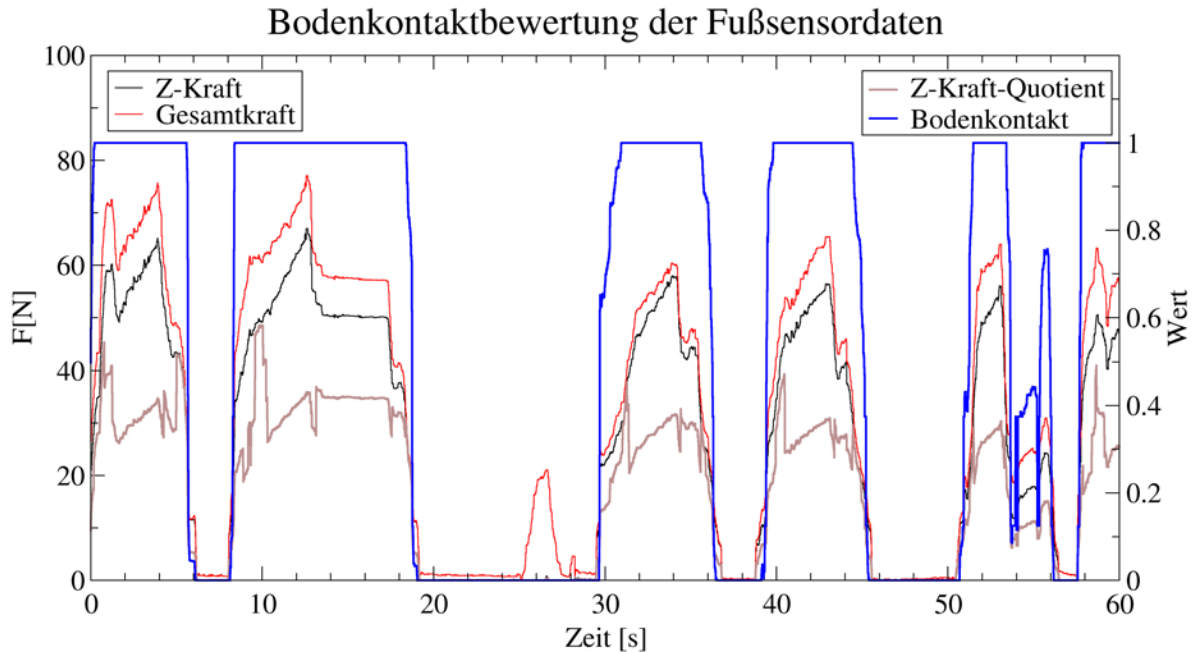


Abbildung 7.3: Bodenkontaktbewertung der Fußsensordaten: Bei normalem Verlauf sind während der Schwingphase alle Kraftkomponenten fast Null (Zeit 7 s), so dass der *Bodenkontakt* zu 0 bestimmt wird. Zu Beginn der Stemmphasen steigen die *z-Kraft*, der *z-Kraft-Quotient* sowie die *Gesamtkraft* schnell an (Zeit 8 s), so dass der *Bodenkontakt* zu 1 wird. Tritt eine Kollision während der Schwingphase auf, steigt die *Gesamtkraft* an (Zeit 26 s); der *Bodenkontakt* wird korrekt zu 0 bestimmt, da die *z-Kraft*-komponente nicht ausschlägt. Beim Auftreten und Abheben des Fußes (Zeit 30 s und 38 s) wird nur teilweiser *Bodenkontakt* festgestellt. Wird ein Bein entlastet, so dass es leicht abrutschen kann, nimmt die Bewertung des *Bodenkontakt* ab (Zeit 55 s). Die Odometrie berücksichtigt die Beinbewegungen nur teilweise.

Wie schon in Abschnitt 3.2.2.3 angedeutet leiden Magnetfeldsensoren insbesondere bei Einsatz auf Robotern unter Störeinflüssen: Elektrische oder magnetische Felder sowie die Ansammlung von Metallen verzerren die Stärke und Richtung des zu messenden Erdmagnetfeldes. Störgrößen lassen sich hierbei durch geeignete Platzierung des Sensors und einer Entzerrung reduzieren: Hierzu wurde LAURON auf einen ansteuerbaren Drehteller fixiert und die verzerrten Sensordaten bei definierten Winkelstellungen während einer vollständigen Umdrehung aufgezeichnet. Bei geeigneter Sensorposition auf LAURON ergab sich eine bijektive Zuordnung, so dass eine eindeutige Umkehrfunktion zur Entzerrung definierbar ist. Die Ergebnisse dieser Entzerrung sind in Abbildung 7.4 dargestellt.

Im Laufe der Kalibrierungs-Experimente stellte sich weiterhin heraus, dass die gyroskopisch-stabilisierten Werten für die Zwecke dieser Arbeit ungeeignet sind: Einerseits benötigen die stabilisierten Werte nach einer Richtungsänderung mehrere Sekunden bis sich ihr Niveau komplett „eingependelt“ hat. Andererseits wird durch die vorgeschaltete Filterung die Grundannahme des eingesetzten Kalman-Filters (weißes Sensorrauschen)

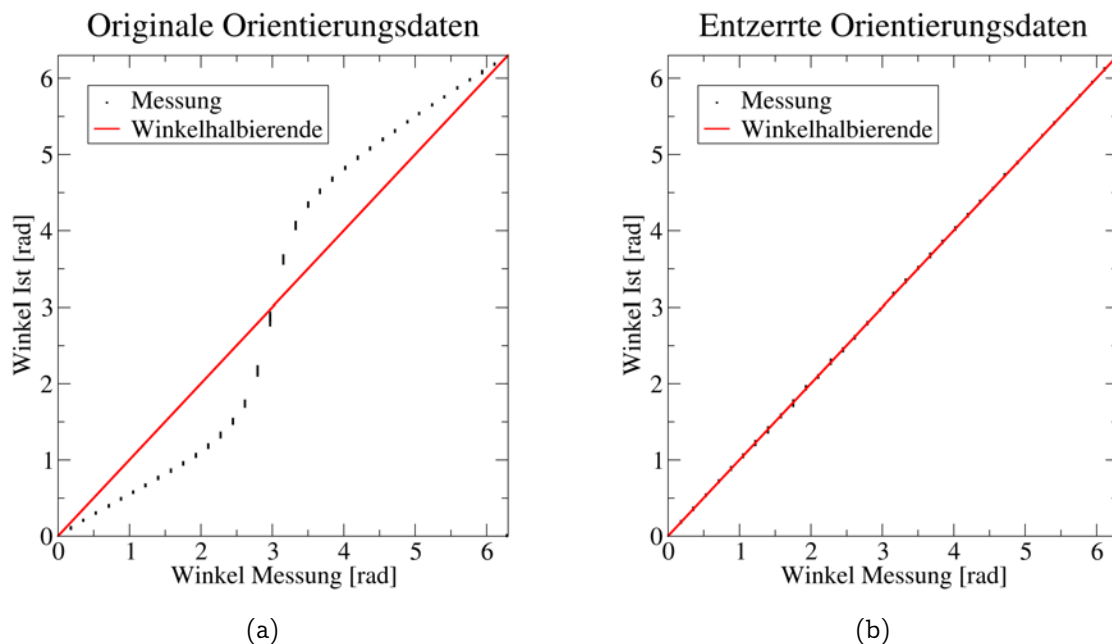


Abbildung 7.4: Kalibrierung der Magnetfeldsensoren: Aufgetragen ist der vom Drehteller eingestellte Ist-Winkel gegenüber dem vom Sensor gemessenen. Der Drehteller wurde jeweils um 10° gedreht und nach einer 5-sekündigen Pause 5 s lang alle 50 ms ein Messwert aufgezeichnet. Hierher rühren die Positionssprünge alle 100 Messungen. Zur Verdeutlichung der Abweichung wurde noch die Winkelhalbierende eingezeichnet, die den Verlauf einer idealen Messung darstellt. (a) Anhand der Abweichungen der Rohdaten wurde eine Funktion zur Entzerrung bestimmt und anschließend das Experiment mit den entzerrten Sensordaten wiederholt. (b) Nach der Entzerrung sind Abweichungen bis auf einen verbleibenden Rauschanteil nicht mehr zu erkennen.

verletzt (Abbildung 7.5). Die erhöhte Position auf dem Rücken von LAURON IV in Abbildung 7.2(b) wurde schließlich als geeignete Sensorposition identifiziert (Haug und Rühl, 2005).

Die vorgeschlagene Entzerrung löst nicht alle Probleme der Magnetfeldsensoren: In der Einsatzumgebung auftretende Störquellen können nicht erfasst werden. Jedoch erstrecken sich externe Störeinflüsse im freien Gelände oft auf eine größere Fläche. Damit wird zwar die absolute Orientierung verfälscht, aber wiederholte Messungen in der lokalen Umgebung kommen zum gleichen Ergebnis. Darüberhinaus wird die Kalibrierung bei Stillstand des Roboters durchgeführt, obgleich die Motoren aktiviert werden können, um die Beinpositionen zu halten. Bei dem realen Laufen werden einerseits die Beine bewegt. Andererseits treten größere Motorbelastungen auf. Weitere Experimente mit frei in der Luft bewegten Beinen zeigen, dass Bewegungsartefakte zu beobachten sind (Abbildung 7.6). Mit deren Auswirkung auf die Orientierungsbestimmung befassen sich die Lokalisationsexperimente in Abschnitt 7.3.

Es bleibt noch festzustellen, dass die beschriebene Kalibrierung nur in der horizontalen Ebene durchgeführt wurde. Bei größerer Neigung des Körpers zeigt sich, dass eine veränderte Roboterkonfiguration auch andersartige Verzerrungen bewirkt. Eine Durchführung dieser Kalibrierung für unterschiedliche Körperneigungen ist daher sinnvoll.

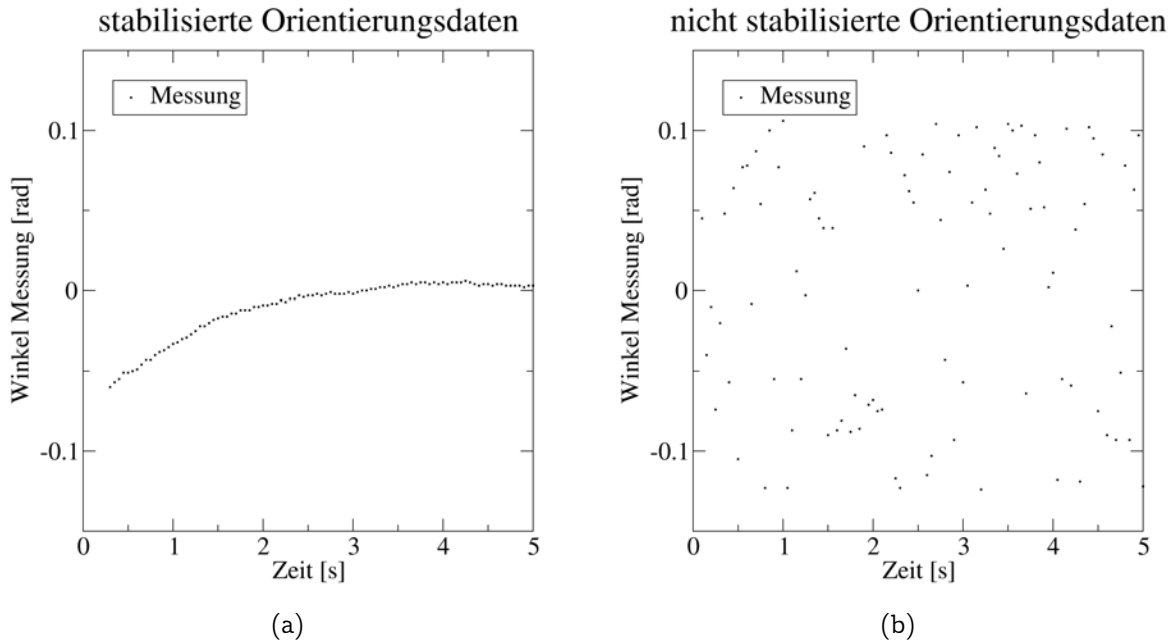


Abbildung 7.5: Stabilisierung der Magnetfeldsensoren: Aufgetragen sind jeweils die 100 Einzelmessungen während eines 5 s Intervalls. (a) Die gyroskopisch-stabilisierten Messwerte benötigen zusätzlich zu den hier nicht aufgezeichneten 3 s Stillstand des Drehtellers in etwa weitere 3 s während der Datenaufzeichnung, um den Stillstand wiederzugeben. (b) Die nicht stabilisierten Messwerte sind zwar weitaus mehr verrauscht, jedoch ist gerade weißes, gaußsches Rauschen eine Grundannahme des Kalman-Filters (vergleiche Abschnitt 3.5). Das Verhalten von vorgefilterten Messdaten ist im Kalman-Filter Systemmodell nicht repräsentiert und führt daher zu suboptimalen Ergebnissen.

7.1.4 GPS

Zur Ermittlung der absoluten Position von Laufmaschinen sind GPS Sensoren sinnvoll. Da teure RTK-Systeme nicht verfügbar waren und die Traglast der Laufmaschine begrenzt ist, wurde ein u-blox-GPS-Empfänger⁴ mit der Möglichkeit des satellitengestützten DGPS-Empfangs zur Navigation angewandt (Zacharias, 2004).

Erste Experimente zeigten, dass eine Antennenposition über dem Zentralkörper für den GPS-Empfang von Vorteil ist. Abbildung 7.2(b) zeigt die aktive GPS-Antenne oben auf der Halterung der Rundumkamera. Der eingesetzte GPS-Empfänger verfügt über mehrere Betriebsmodi: Zum einen kann die Nutzung von SBAS, dem satellitengestützten DGPS, an- und abgeschaltet werden. Zum anderen erlaubt der Empfänger die Konfiguration unterschiedlicher Bewegungsmodelle wie *stationär*, *Fußgänger* oder *Automobil*. Nach Experimenten auf LAURON hat sich gezeigt, dass das Bewegungsmodell *Fußgänger* für LAURON der geeignetste ist. Wie zu erwarten, werden die Positionsdaten beim aktiviertem SBAS genauer (Haug und Rühl, 2005).

Die Versuche zeigten aber auch, dass der Empfang der SBAS-Korrekturen oft lückenhaft ist, was die Verlässlichkeit der GPS-Positionsdaten stark reduziert.

⁴<http://www.u-blox.com>

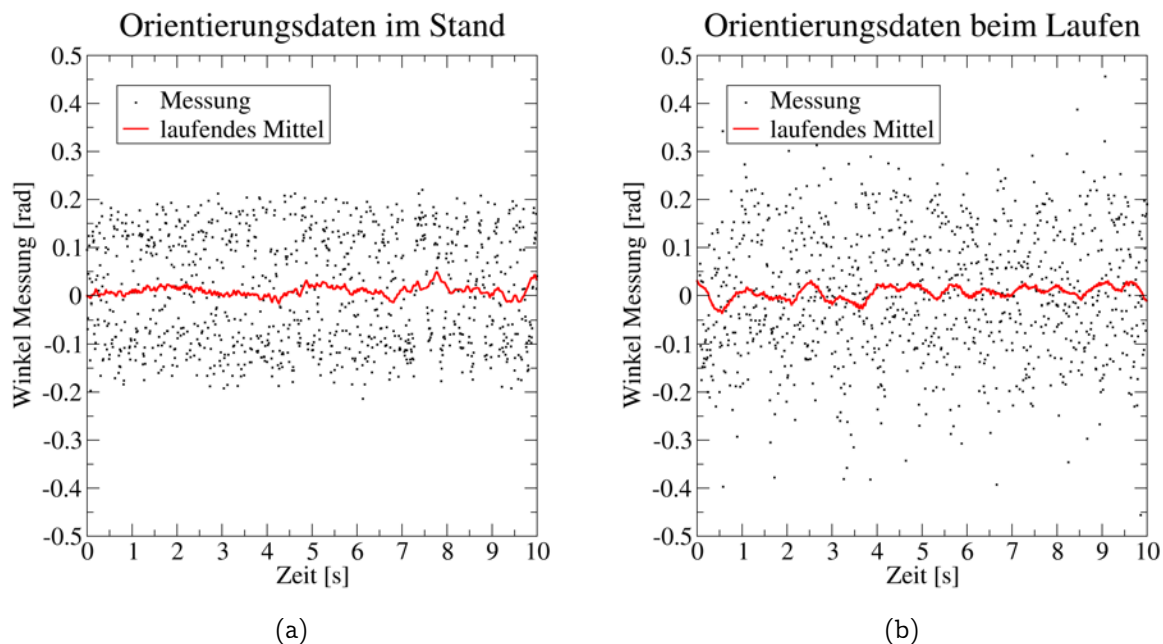


Abbildung 7.6: Einfluss des Laufens auf die Magnetfeldsensoren: Aufgetragen sind jeweils die Einzelmessungen während eines 10s Intervalls sowie ein laufendes Mittel über der Zeit. Im Stand (a) ist das Rauschen der Messwerte geringer als beim Laufen (b). Kann dem laufenden Mittel im Stand (a) ein so genannter Random-Walk unterstellt werden, fällt ein zyklisches Auf und Ab während des Laufens (b) ins Auge. Hierbei handelt es sich offensichtlich um Artefakte der zyklischen Laufbewegungen.

7.1.5 Laserabstandssensor

Für die Experimente stand als optischer Abstandssensor ein kompakter ($50 \times 50 \times 70$ mm), leichter (160 g) und stromsparender (2,5 Watt) Linienlasers zur Verfügung, der durch Neigen des Roboterkopfes die Umwelt Zeile für Zeile abtastet. Der HOKUYO URG-X003⁵ ist ein Infrarot-Laser Abstandssensor (Wellenlänge 785 nm) mit bis zu 240° Öffnungswinkel bei einer Winkelauflösung von $0,36^\circ$ und bis zu 4 m Reichweite. Die Genauigkeit ist mit 1 mm angegeben. Alle 100 ms kann ein kompletter Linienscan mit 683 Distanzwerten geliefert werden (Abbildung 7.2(d)).

Zur Koordinatentransformation der Messpunkte in das Roboterkoordinatensystem wird auf die aktuelle Sensorkopforientierung zurückgegriffen. Bei fehlerhafter Synchronisation dieser Informationen wird das Messergebnis verfälscht, weshalb eine vierstufige Messstrategie eingeführt wurde: Planen – Sensorkopf Bewegen – Sensorkopf Anhalten – Messdaten Aufnehmen. Hierdurch ist es möglich, gezielt Lücken im Umweltmodell zu füllen und den Anforderungen von Bahnplanung und Adaption nachzukommen. Mit Hilfe der aktuellen globalen Roboterposition, die von der Lokalisationskomponente zur Verfügung gestellt wird, lassen sich die absoluten Koordinaten der gemessenen Punkte im Raum bestimmen und schließlich in das Umweltmodell eintragen. Auf ähnliche Art und Weise werden auch die Fußaufsetzpunkte behandelt, so dass deren globale Position dem Umweltmodell zur Verfügung gestellt werden können.

⁵<http://www.hokuyo-aut.jp/>

7.2 Reaktive Laufmaschinensteuerung

Die Steuerung der sechsbeinigen Laufmaschine LAURON IV basiert auf dem in C++ geschriebenen Software-Framework MCA2 (Scholl et al., 2001). Sie erlaubt reaktives Durchqueren von unstrukturiertem Gelände (Gaßmann, 2000). Im Folgenden werden die für die Umsetzung der Navigationsalgorithmen relevanten Teile kurz vorgestellt:

Gelenkregelung und Sensordatenerfassung finden auf den hardwarenahen UCoM-Einheiten statt. Die teilweise vorgefilterten Sensordaten werden über CAN-Bus an die zentrale Rechneinheit übermittelt.

Sensordatenverarbeitung und -konvertierung ist die Aufgabe der unteren Steuerungsebene auf dem Kompakt-PC: Die Sensordaten werden empfangen und gefiltert (Medianfilter, Durchschnittsfilter), skaliert oder konvertiert (Zählpulse in Winkel wandeln). Des Weiteren werden diese abgeleitet, synchronisiert oder wie die Orientierungsdaten entzerrt. Komplexere Auswertungen der Sensordaten wie eine Bodenkontaktbewertung oder eine Kollisionsbewertung werden ebenfalls an dieser Stelle durchgeführt. Dieser von der speziellen Hardware abstrahierenden Ebene ist darüber hinaus die Berechnung der direkten und indirekten Kinematik zuzuordnen.

Die **Einzelbeinsteuerung** generiert die Trajektorie eines Fußes entsprechend der aktuellen Beinphase. In der Stemmphase wird der Fuß am Boden nach hinten gedrückt, um in der Schwingphase in einem Bogen durch die Luft wieder nach vorne gesetzt zu werden. Die Einzelbeintrajektorie ist recht flexibel gehalten und ermöglicht einfache Basisreaktionen auf unvorhergesehene Ereignisse wie beispielsweise einer Kollision des Beines während der Schwingphase.

An dieser Stelle werden die von den Planungsebenen, Bahnplanung und Adaption, vorgegebenen Steuerungsparameter *Schrittlänge*, *Belastungsfaktor*, *Schrittbreite* und *-höhe* in der reaktiven Laufmaschinensteuerung verarbeitet. Sie beeinflussen die Ausprägung der Einzelbeintrajektorie. Um die Fußaufsetzpunktplanung zu realisieren, musste die Einzelbeinsteuerung erweitert werden: Vor dem Beginn der Schwingphase berechnet diese zunächst den geplanten Fußaufsetzpunkt und wartet mit dem Schwingphasenstart bis die Planung einen geeigneten Aufsetzpunkt gefunden und einen entsprechenden *Offset* bereitgestellt hat.

Die **Beinkoordination** sorgt als zentrale Instanz für einen geordneten Ablauf der sechs Beine. Unter Berücksichtigung globaler Aspekte wie der Stabilität oder der Vermeidung einer Kollision von Beinen untereinander, versucht diese, die vorgegebene Gangart einzuhalten. Durch den Steuerungsparameter *Belastungsfaktor* werden Phasenverschiebungen der Beine untereinander definiert. Die Vorgabe der *Zykluszeit* bestimmt die Geschwindigkeit, mit der die Beinphasen voranschreiten.

Die **Haltungskontrolle** achtet als übergeordnete Instanz auf eine angemessene Körperhaltung. Grundsätzlich wird versucht, den Steuerungsvorgaben in Bezug auf *Körperhöhe*, *-position* und *-neigung* Folge zu leisten.

Die reaktive Steuerung erlaubt der Laufmaschine das Übersteigen von Hindernissen und das Durchqueren von unstrukturiertem Gelände. Schwierige Hindernisse können oftmals durch geschicktes Vorgehen einfacher überwunden oder gar umgangen werden. Eine Vorausschau um die nächsten paar Schritte wird jedoch erst durch die realisierten Navigationsmethoden möglich. LAURON IV ist robust gegenüber kleineren Planungsungenauigkeiten, die reaktive Steuerung kann die Ungenauigkeiten ausgleichen.

7.3 Lokalisation

Der in Abschnitt 3.8 skizzierte Lokalisationsalgorithmus wurde implementiert und dessen Leistungsfähigkeit auf LAURON IV getestet. Die Grundlage der Odometrie berechnung bilden die hierzu aus den Fußkraftsensoren ermittelten Gewichte. Die Daten zur Orientierungskorrektur liefert der Orientierungssensor. Der GPS-Empfänger stellt die Positionskorrekturdaten zur Verfügung.

Zu Testzwecken wurde als Experimentparcours ein definiertes Rechteck vorgegeben, das abzulaufen war. Hierzu wurde LAURON IV manuell auf dem Rechteck entlangmanövriert. Ein geradliniges Ablaufen des Rechtecks wird aber durch die Tatsache erschwert, dass sich die Laufmaschine in der Regel auch bei der Sollvorgabe *geradeaus* durch innere Verspannungen, Fertigungs-, Kalibrierungsungenauigkeiten, usw. immer in eine Richtung wegdreht. Abhängig von der Laufgeschwindigkeit, der Beobachtungsgabe und der Reaktionszeit des Bedieners ergeben sich hierdurch Abweichungen von einem Rechteck, die erhebliche Beträge annehmen können. Die Laufmaschine wurde an den Eckpunkten wieder korrekt ausgerichtet, bevor die nächste Seite des Rechtecks abgeschrieben wurde, so dass die Ergebnisse hierdurch nicht an Aussagekraft verlieren.

Es sei anzumerken, dass bei der Präsentation der folgenden Experimente darauf geachtet wurde, repräsentative Messungen wiederzugeben. Es wurden viele Messreihen mit wechselnden Abweichungen durchgeführt. Mit einem kalibrierten Robotersystem lassen sich die folgenden Ausführungen bestätigen.

7.3.1 Odometrie

Um die Güte der Odometrie zu testen, wurden die ersten Laufexperimente auf ebenem Fußboden bei langsamer Fortbewegungsgeschwindigkeit durchgeführt. Dies reduziert das Rutschen der Beine auf ein Minimum. Es wurde ein Quadrat mit Seitenlänge von 1,8 m abgelaufen. Abbildung 7.7 zeigt die während des Testlaufs aufgenommenen Daten.

An der Projektion des Steckenverlaufs in die xy -Ebene (Abbildung 7.7(a)) lässt sich die Güte der Lokalisation ablesen. Im Idealfall müsste das Quadrat geschlossen sein bzw. die Linien der zweifach beschrittenen Seite aufeinanderliegen. Der Roboter „verfehlt“ den Startpunkt um etwa 40 cm. Der resultierende Orientierungsfehler divergiert nach einer Umrundung zu 20° . Die durchschnittlich gemessene Seitenlänge beträgt 1,88 m, was einem Messfehler von 4,8% in Längsrichtung entspricht. Der durchschnittlich gemessene Winkel beträgt 95° , was ein Messfehler von 5,6% ergibt. Es ist festzustellen, dass sämtliche Fehler ein positives Vorzeichen aufweisen, was durch leichtes Rutschen der Beine

beispielsweise beim Lastwechsel erklärbar ist.

Abbildung 7.7(b) zeigt den Winkelverlauf der beiden Neigungswinkel *roll* und *pitch* zum Untergrund. Diese wandern in einem Bereich von bis zu 5° ($0,87$ rad), obwohl beide Orientierungskomponenten bis auf durch das Laufen bedingtes Hin- und Herwiegen bei konstant 0° lagen. Die zu beobachtenden steilen An- und Abstiege sind hingegen durch die Änderung des *yaw*-Winkels an den Ecken des Quadrates bedingt. Bei einer 90° -Drehung vertauschen sich die Beträge der beiden Komponenten *roll* und *pitch* gerade, weil sie den Winkel zum globalen, fixen Koordinatensystem angeben. Ohne Korrekturen können diese Fehler divergieren und somit auch die Längenmessung verfälschen.

Schließlich zeigt Abbildung 7.7(c) den Verlauf der gemessenen Körperhöhe. Auch diese war während der gesamten Versuchsdauer im Grunde konstant. Da die Koordinaten zu Beginn des Versuches auf 0 zurückgesetzt wurden wäre die Nulllinie hier der korrekte Wert. Aber es ist ein konstanter Anstieg des Körpers auf 47 cm zu verzeichnen, was einer Steigung von 0.374° entspricht. Die Ursachen hierfür sind vor allem in der nicht exakten Bestimmung des Fußaufsetzzeitpunktes sowie mechanischen Ungenauigkeiten begründet. In früheren Versuchen war dieser systematische Fehler, dessen Vorzeichen sich beispielsweise durch einen Gangartwechsel auch ändern kann, wesentlich größer. Erst ein Ignorieren der zurückgelegten Wegstrecke direkt beim Aufsetzen eines Fußes führte zu besseren Ergebnissen. Hieran wird deutlich, wie wichtig es ist, den Aufsetzzeitpunkt des Fußes exakt zu bestimmen.

7.3.2 Orientierungskorrekturen

Während des obigen Laufexperimentes wurden die Odometriedaten in einem unabhängig laufenden Filtermodul gleichzeitig mit absoluten Orientierungsdaten des Orientierungssensors korrigiert. Die Abbildung 7.8 stellt die Ergebnisse in entsprechender Form grafisch dar.

Auch bei Korrektur der Orientierungswerte wird das Quadrat nicht geschlossen. Es ist ein Versatz von 47 cm zu beobachten (Abbildung 7.8(a)). Der resultierende Orientierungsfehler beträgt hingegen 0° . Nach einer Runde weist der Roboter die gleiche Orientierung auf als zu Beginn. Dies ist dem absoluten Messcharakter des Orientierungssensors zu verdanken: kurzzeitige Fehler kumulieren nicht über der Zeit, sie können korrigiert werden. Die Messfehler bzgl. der zurückgelegten Distanzen verhalten sich hingegen ähnlich dem Odometrieexperiment.

Die Trapezform des gemessenen quadratischen Streckenverlaufs ist auf Verzerrungen im Magnetfeld zurückzuführen. Beim Laufen in Richtung der positiven bzw. negativen *y*-Achse ist eine konstante Verzerrung der *yaw*-Orientierungskomponente von etwa 6° – 7° zu beobachten.

Die beiden anderen Orientierungskomponenten spiegeln hingegen den realen Verlauf der Roboterorientierung während des Experimentes wider: Eine von Bewegungsruschen überlagerte Nulllinie. Die Fehler der Odometrieberechnung können vollständig korrigiert werden (Abbildung 7.8(b)). Die korrekten Neigungswinkel des Zentralkörpers bewirken weiterhin, dass der Höhenverlauf unverzerrt wiedergegeben wird. Der Anstieg der Körperhöhe stellt nun eine Gerade dar (Abbildung 7.8(c)). Bei jedem Lastwechsel

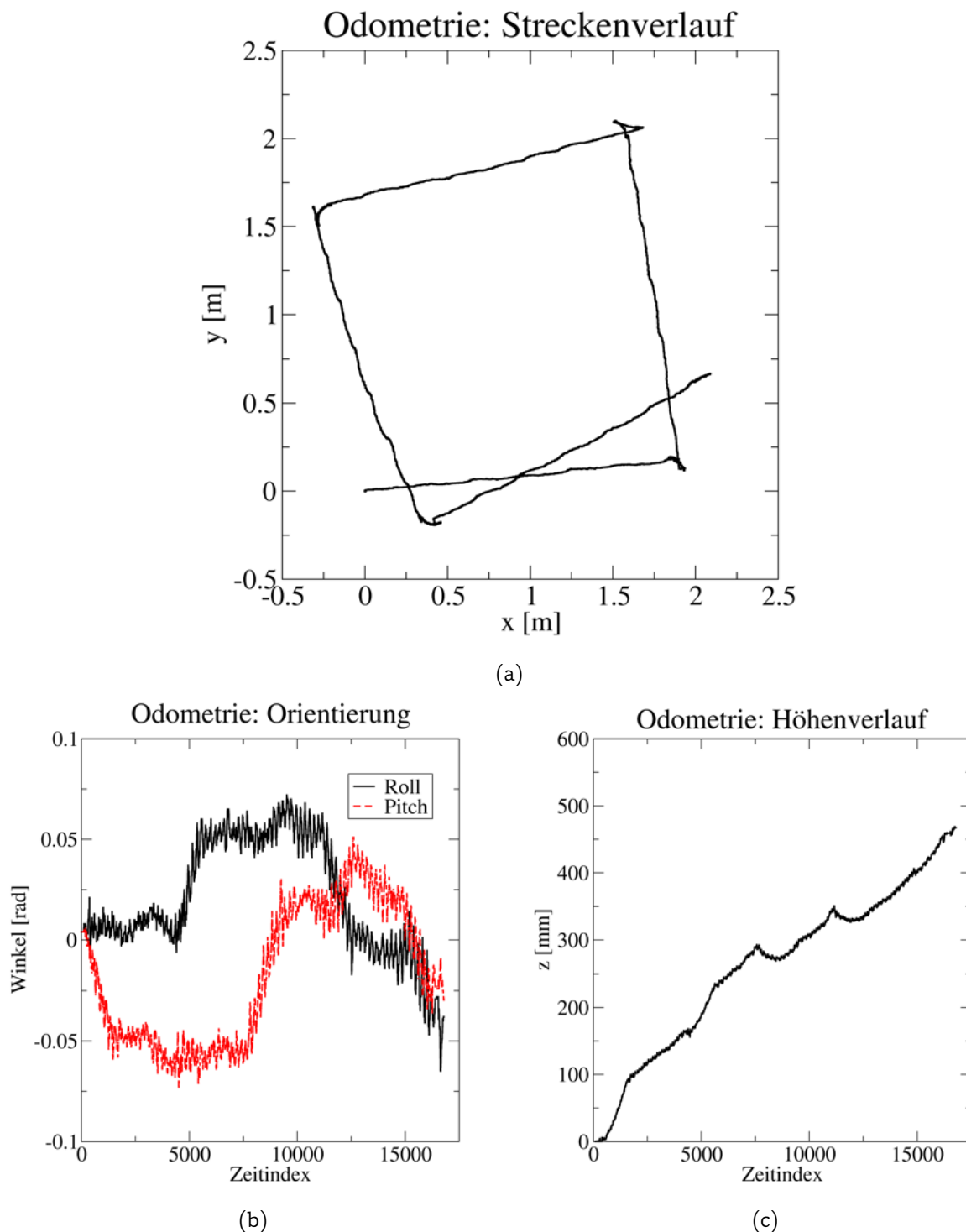


Abbildung 7.7: Odometriexperimente: (a) Der Streckenverlauf der Odometriemessung beim Umrunden eines flachen Quadrates mit 1,8 m Seitenlänge. Die Odometrie überschätzt die zurückgelegten Wegstrecken und -winkel. Die Fehler akkumulieren sich. (b) Der Verlauf der Winkel *roll* und *pitch* bezüglich der Erdoberfläche ähnelt einem so genannten Random-Walk. Durch Rauschen verursachte Orientierungsänderungen können sich zu zu einem unbegrenzt wachsenden Fehler aufsummieren. (c) Beim Höhenverlauf über der Zeit ist das Phänomen eines systematischen Fehlers zu beobachten. Gegen Ende des Versuchs lag der Zentralkörper um 47 cm höher als zu Beginn.

der Beine erfuhr die Zentralkörperhöhe in diesem Experiment einen Fehler der Größenordnung 0,7 mm, der sich schließlich zu einer Endhöhe von 57 cm aufsummierte.

7.3.3 Positionskorrekturen

Zur Beurteilung von GPS-Positionskorrekturen hatte die Laufmaschine ein Rechteck der Größe $7,5\text{ m} \times 3,5\text{ m}$ auf einem mit Steinplatten belegten Boden zu umrunden. Abbildung 7.9 stellt die für Vergleichszwecke parallel berechneten Positionsdaten gegenüber: Odometrie, Odometrie mit Orientierungskorrekturen, Odometrie mit GPS-Positions- und Orientierungskorrekturen sowie schließlich die empfangenen GPS-Rohdaten.

Die GPS-Rohdaten streuen sehr weit in einem Bereich von $20\text{ m} \times 15\text{ m}$. Unter Berücksichtigung der gelaufenen Strecke verbleiben Positionsungenauigkeiten von über 10 m, obgleich bei Einsatz von SBAS-DGPS Positionsungenauigkeiten von 1–3 Metern erwartet wurden (vergleiche hierzu Tabelle 3.1). Die Metadaten des GPS-Empfänger bestätigen, dass DGPS-Korrekturen nur für kurze Zeitabschnitte zur Verfügung standen. Das abgelaufene Rechteck ist im Fusionsergebnis des Kalman-Filters nicht erkennbar.

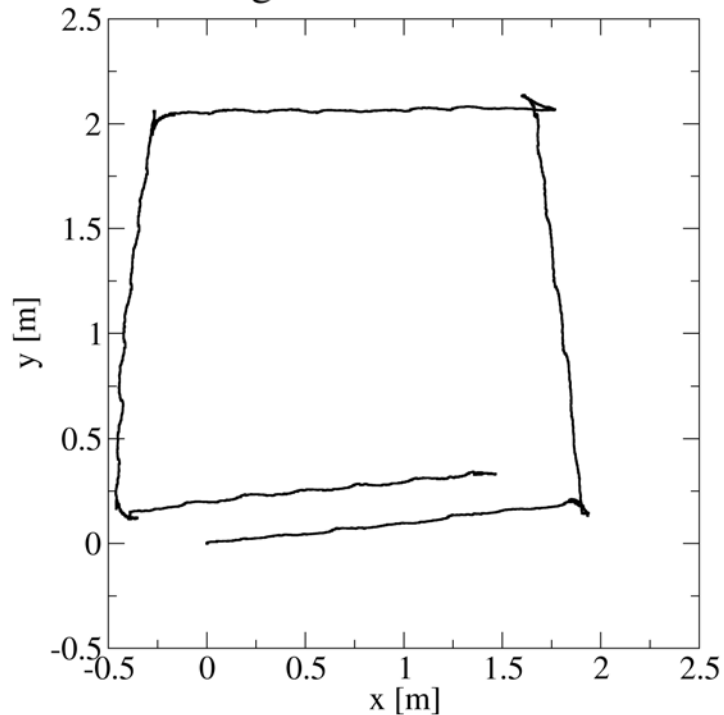
Ein weiterer Versuchslauf fand in einem Kiesbett der Größe $4,5\text{ m} \times 3,5\text{ m}$ statt. Hierbei schwankten die GPS-Daten während der Messung um 15 m. Die abgelaufenen Rechteckseiten sind im Fusionsergebnis nicht mehr zu erkennen (Abbildung 7.10). Für die gemessenen GPS-Höhen gilt Vergleichbares: die Streuung während der Versuche lag ebenfalls bei bis zu 15 m (Abbildung 7.11).

Dies lässt den Schluss zu, dass GPS-Daten in dieser Form zur Korrektur der Laufmaschinenposition nicht herangezogen werden können. Man könnte die Experimente an anderer Stelle wiederholen, um die Ergebnisse bei verfügbaren SBAS-DGPS-Korrekturen zu verbessern; autonomes Laufen in beliebigem Gelände ist aber nicht möglich. Ein genauere GPS-Empfänger mit RTK-Fähigkeiten oder der Aufbau eines lokalen DGPS mit eigener Basisstation kann zur Verbesserung der Genauigkeiten beitragen.

Betrachtet man die reinen Odometriedaten auf den Steinplatten in Abbildung 7.9, so kann ein durchschnittlicher Distanzfehler von etwa 9,25% festgestellt werden; die Orientierungsungenauigkeiten sind ebenfalls um eine Größenordnung gestiegen. Hierfür ist die schnellere Fortbewegungsgeschwindigkeit des Roboters im Vergleich zu den Odometrieexperimenten aus Abschnitt 7.3.2 verantwortlich. Offensichtlich hatte LAURON IV im Kies durch das Einsinken der Fußspitzen einen weitaus besseren Stand als auf den Steinplatten. Der Distanzfehler der Odometrie in Abbildung 7.10 beträgt nur 4,4%, ein Winkelfehler ist kaum auszumachen.

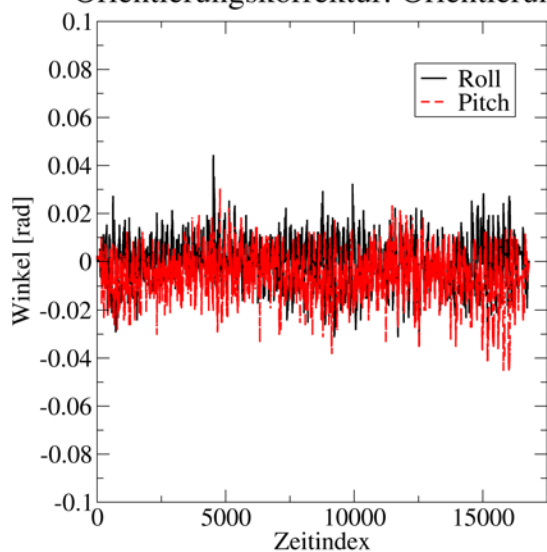
Die erhöhte Fortbewegungsgeschwindigkeit des Roboters verstärkt die durch die Motoren erzeugten Störungen im Magnetfeld; die Winkelverzerrung der Magnetfeldsensoren steigt auf 15° – 18° an. In den abgebildeten Streckenverläufen mit aktivierten Orientierungskorrekturen, kann wiederum eine Art Trapezform beobachtet werden (Abbildungen 7.9 und 7.10). Die resultierenden Höhen von 48 cm auf den Steinplatten und 75 cm im Kies bewegen sich im schon zuvor beobachteten Rahmen (Abbildungen 7.11(a) und 7.11(b)).

Orientierungskorrektur: Streckenverlauf



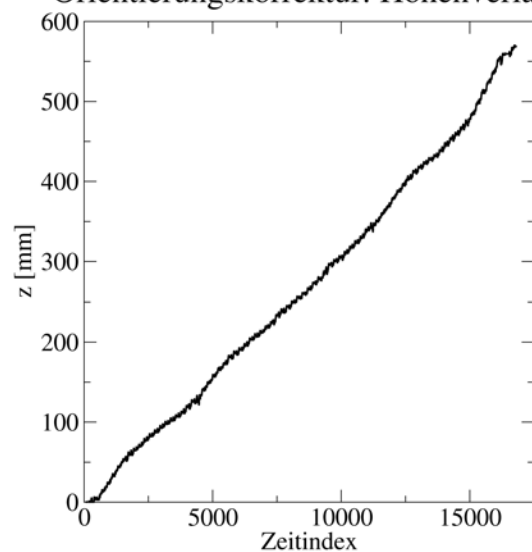
(a)

Orientierungskorrektur: Orientierung



(b)

Orientierungskorrektur: Höhenverlauf



(c)

Abbildung 7.8: Experimente zur Orientierungskorrektur: (a) Anstatt eines Quadrates wird unter Einbezug der Orientierungskorrekturen der Verlauf eines Trapez wiedergegeben. Die *yaw*-Komponente bleibt trotz Kalibrierung der Magnetfeldsensoren verzerrt. (b) Die Orientierungsfehler der Odometrieberechnung können durch den absoluten Orientierungssensor vollständig korrigiert werden. (c) Aufgrund der korrekt berücksichtigten Roboterneigung wird der Höhenverlauf geglättet. Der Verdacht auf einen systematischen Fehler bei der Höhenberechnung erhärtet sich.

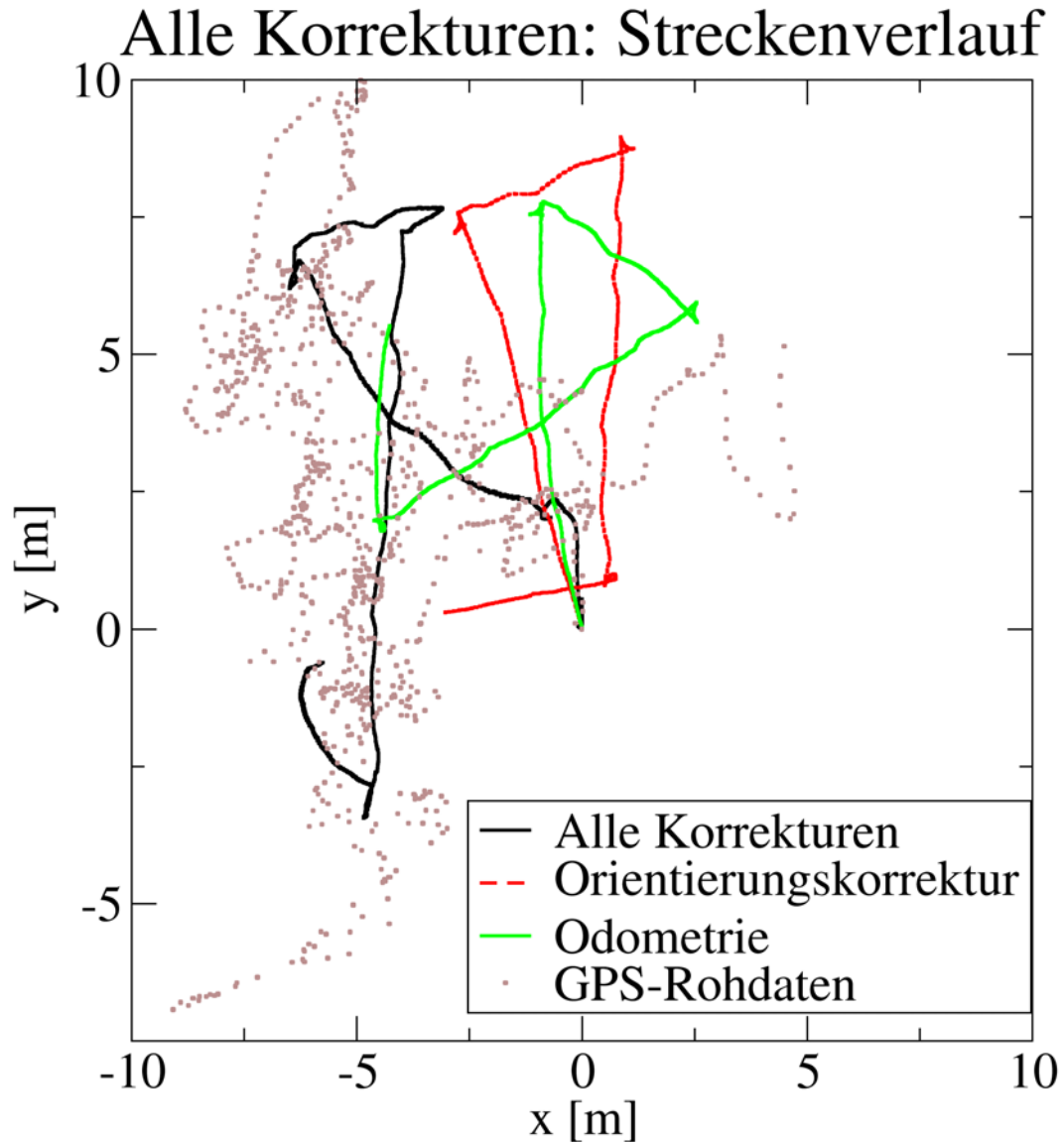


Abbildung 7.9: Experimente zur Positions- und Orientierungskorrektur im Freien: Laufen im flachen Rechteck der Größe $7,5 \text{ m} \times 3,5 \text{ m}$ auf Steinplatten. Die Odometrie erfährt durch die höhere Laufgeschwindigkeit größere Fehler. Aus gleichem Grund nimmt auch die Verzerrung des Magnetfeldes bei den orientierungskorrigierten Daten zu. Die GPS-korrigierten Daten sind zur Lokalisation von LAURON nicht zu gebrauchen. Dies ist u.a. durch die nur selten verfügbaren SBAS-DGPS-Korrekturen begründet.

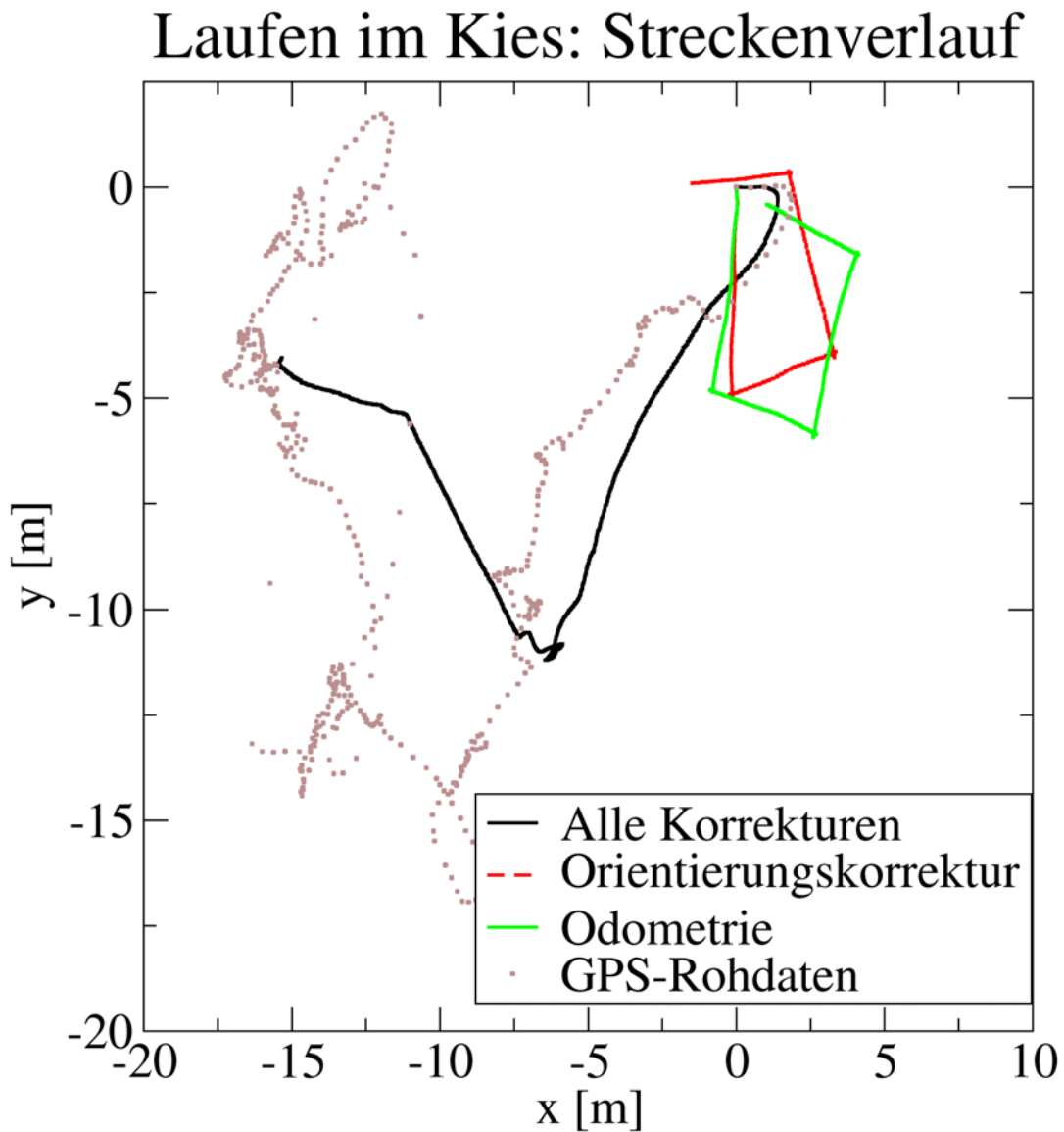


Abbildung 7.10: Experimente zur Positions- und Orientierungskorrektur in Kies. Durchschreiten eines Rechtecks der Größe $4,5\text{ m} \times 3,5\text{ m}$ im Kiesbett. LAURON IV findet im Kies besseren Halt als auf den Steinplatten und erzielt sehr gute Odometrieergebnisse. Die Magnetfeldverzerrungen sind identisch mit den auf den Steinplatten beobachteten. Die GPS-Positionsdaten sind hingegen nochmals ungenauer.

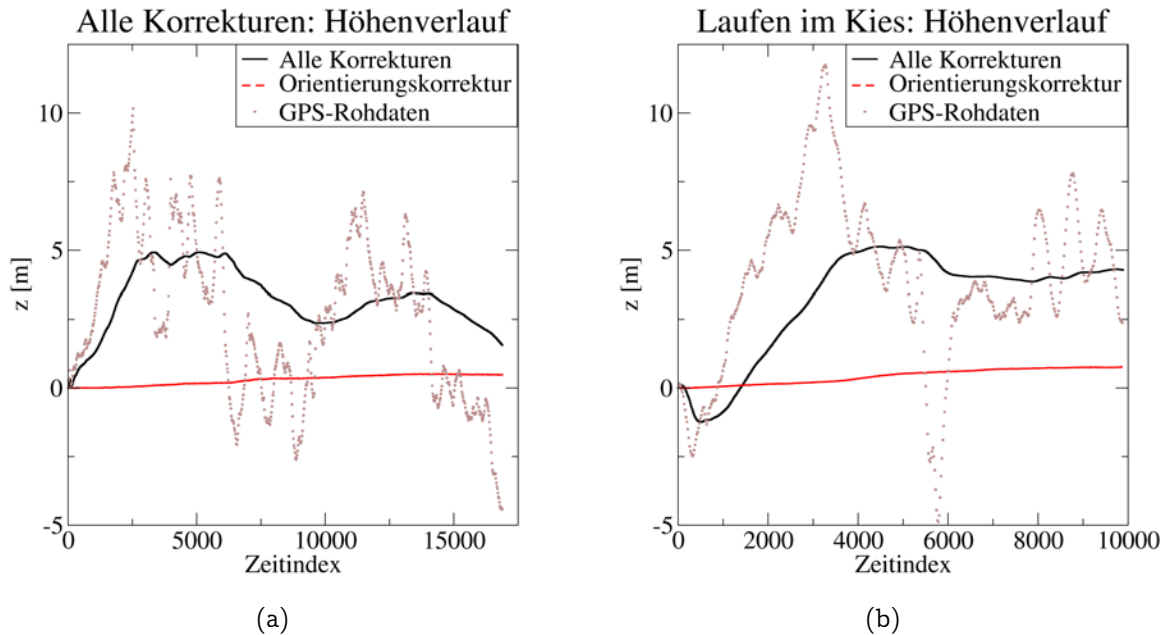


Abbildung 7.11: Höhendaten bei Positions- und Orientierungskorrektur. (a) Die Höhendaten der GPS-Korrekturen schwanken um bis zu 15 m. Ohne Positionskorrekturen bleiben sie im bisher bekannten Rahmen. (b) Beim Laufen im Kies weichen die Höhendaten sowohl mit als auch ohne Positionskorrektur nur marginal von denen auf den Steinplatten ab.

7.3.4 Zusammenfassung und Erkenntnisse

Die aufgeführten Experimente zeigen, dass die Odometrieberechnung mit der vorgestellten Gewichtung der Beine auf unterschiedlichen Untergründen stabil arbeitet. Je nach Laufgeschwindigkeit und Untergrund liegen die Ungenauigkeiten bei 5–10%. Da die Fehler vor allem auf rutschende Beine zurückzuführen sind, werden die zurückgelegten Distanzen und überstrichenen Winkel immer überschätzt. Da kumulierte Winkelfehler die Position auch bei zukünftig korrekten Messungen verfälschen, kann die Odometrie alleine nur für kurze Strecken eingesetzt werden.

Die Fusion absoluter Orientierungsdaten schafft hier Abhilfe. Durch verlässliche Information über die absolute Roboterneigung zum Untergrund können durch ungenaue Odometrie bedingte Neigungsfehler korrigiert werden. Stünden korrekte absolute Orientierungsdaten auch für die *yaw*-Komponente zur Verfügung, würde das konzeptionierte Kalman-Filter auch derartige Fehler ausnahmslos korrigieren.

Jedoch erweist sich der Einsatz von Magnetfeldsensoren auf LAURON IV nicht als die beste Lösung. Zwar können durch Kalibrierung statische Verzerrungen des Magnetfeldes herausgerechnet werden, von der Laufbewegung sowie von der -geschwindigkeit abhängige Verzerrungen sind jedoch nicht so einfach zu korrigieren. Eine sich dynamisch an die Laufgeschwindigkeit anpassende Fehlerkompensation stellt einen theoretischen Lösungsansatz dar. Berücksichtigt man aber die Ausführungen nach Abschnitt 7.1.3, nachdem eine Kalibrierung grundsätzlich auch bei unterschiedlichen Roboterneigungen durchzuführen wäre, ist deren praktische Umsetzung kaum zu bewerkstelligen: Eine vollständige

Kalibrierungsdatenbank würde demnach sowohl für mehrere Robotergeschwindigkeiten als auch für verschiedene Roboterneigungen einen Satz von Korrekturverläufen benötigen. Die Problematik externer Störquellen bliebe in jedem Fall weiterhin ungelöst.

Als weitere Lösung bietet sich der Einsatz einer hochgenauen Sensorik an. Hochgenaue faseroptische Kreisel wie sie u.a. in Flugzeugen zum Einsatz kommen weisen eine geringe Drift auf, dass bei deren Kalibrierung die Geschwindigkeit der Erddrehung am Einsatzort mitbestimmt wird. Auf LAURON IV würde ein solcher Kreisel genügen, um die Orientierung zumindest für den Einsatzzeitraum von einer Stunde genau zu halten.

Ähnlich verhält es sich auch mit den GPS-Positionsdaten. Der für diese Arbeit zur Verfügung stehende kostengünstige GPS-Empfänger verfehlt die von Laufmaschinen geforderte Genauigkeit deutlich. Ungünstige Satellitenkonstellationen, auftretende Mehrfachreflexionen sowie nicht ständig verfügbare SBAS-Korrekturen sind als deren Hauptursache zu nennen. Die um einen Faktor 5–10 genaueren SBAS-DGPS-Daten würden immerhin genügen, das von der Navigationsaufgabe vorgegebene Ziel nicht zu verfehlen. Eine wesentliche Anforderung an die Lokalisation ist die Forderung nach möglichst wenig Positionssprüngen, um den Aufbau des lokalen Umweltmodells nicht zu gefährden. Hierzu sind auch SBAS-DGPS-Daten noch zu ungenau. Der Einsatz eines RTK-GPS-Empfängers mit absoluter Genauigkeit im Zentimeterbereich stellt hierfür die (theoretische) Lösung dar.

Zusammenfassend gilt somit die folgende Aussage: Das konzipierte Kalman-Filter zur Fusion der Odometrie mit absoluten Orientierungs- und Positionsdaten liefert korrekte Ergebnisse. Eine deutliche Verbesserung der Sensorik zur absoluten Orientierungs- und Positionsbestimmung ist hierfür jedoch notwendig.

7.4 Umweltmodellierung für Laufmaschinen

Den Erkenntnissen des letzten Abschnitts folgend bietet die Lokalisation für kurze Strecken verlässliche Ergebnisse, um ein Umweltmodell im Nahbereich aufzubauen. Dieser Abschnitt beschreibt die Umsetzung des in Kapitel 4 konzipierten Modells einer dreidimensionalen geometrischen Urkarte in Form eines erweiterten Inferenz-Gitters auf LAURON. Anschließend wird anhand eines Kartierungsexperimentes die Funktionstüchtigkeit des entwickelten Verfahrens unter Beweis gestellt.

7.4.1 Dimensionierung der Karten

Zunächst stellt sich die Frage nach der Auflösung des über die Umwelt gelegten Gitters, also der Größe der Gitterzellen. Eine sinnvolle Wahl muss in erster Linie die Abmessungen des Roboters berücksichtigen. Unabhängig davon ist es sinnvoll, das Gitter bzgl. der Höhenachse (z -Achse) feiner zu granulieren. Auch der Mensch nimmt Höheninformationen empfindlicher wahr als Ausdehnungen in xy -Richtung. Für den Roboter sind Ausdehnungen in z -Richtung gerade für das zielgerichtete Aufsetzen relevant, da sie die erforderliche Schritthöhe bestimmen. Die Granulierung in der xy -Ebene muss dagegen nicht so fein gewählt werden. Hier bietet sich die Ausdehnung der Füße der Maschine als

Maß an, damit jede Zelle genau einen möglichen Fußaufsetzpunkt beschreibt. In dieser Arbeit wurde daher ausgehend von der Größe der Füße von LAURON eine Gitterzellengröße von $50 \times 50 \times 12,5$ mm gewählt. Die Gitterzelle der globalen $2\frac{1}{2}$ D-Karte weist somit die gleiche Größe auf.

Neben der Auflösung des Gitters ist auch dessen Ausdehnung von Bedeutung. Die Urkarte soll die unmittelbare Umgebung des Roboters widerspiegeln. Auf Grund der langsamen Fortbewegungsgeschwindigkeit und der maximalen Reichweite des Laserscanners von vier Metern wurde eine horizontale Ausdehnung von 10 m konzipiert, um den notwendigen Speicherverbrauch noch in Grenzen zu halten. Dies entspricht 5 m in jeder Bewegungsrichtung, sofern die Laufmaschine sich in der Kartenmitte befindet. Durch die Verwendung eines Octrees zur Speicherung des Gitters mit gleicher Anzahl Zellen in jeder Dimension, ergibt sich somit eine Ausdehnung in der Höhe von 2,5 m. Die Ausdehnung der Höhenkarte ist weniger kritisch, da deren Speicherverbrauch nur dann ansteigt, wenn LAURON die entsprechenden Gebiete auch durchlaufen hat. Sie wird deshalb auf $100 \text{ m} \times 100 \text{ m}$ festgelegt.

7.4.2 Virtuelle Sensoren

In Abschnitt 7.1 wurde die zur Umwelterfassung zur Verfügung stehende Sensorik vorgestellt. Die Auftrittsstelle des Fußes sowie die Abstandsmessungen durch den Laserscanner liefern Daten über die Geometrien in der näheren Umgebung. Neben den direkt durch die Sensoren wahrgenommenen belegten Zellen erhält man bei einer Messung auch implizite Information über nicht belegte Zellen: An der Stelle, an der der Fuß selbst sich befindet, sind keine weiteren Zellen zu finden. Auch der Strahl zwischen Lasersensor und gemessenem Punkt kann als frei deklariert werden (Abbildung 7.12). Somit kann ein virtueller Sensor entworfen werden, der die freien Zellen als solche ins Weltmodell mit einträgt. Aufgrund der relativ grob ermittelten virtuellen Sensorlinie wird dem virtuellen Sensor jedoch eine verminderte Sensorqualität zugesprochen. In der Implementierung auf LAURON wurde für die Fußkontaktpunkte eine Sensorqualität von 0,5, für die Laserabstandsdaten 0,8, sowie für den virtuellen Sensor ein Wert von 0,3 gewählt. Diese Sensorqualitäten gehen als Expertenwissen in den Bewertungsprozess ein.

Das Einführen des virtuellen Sensors hat jedoch zur Folge, dass sich der Berechnungsaufwand zum Eintragen insbesondere eines einzelnen Laserabstandswertes erhöht. Bei einer Messung in 2 m Entfernung ist eine Zelle als belegt zu klassifizieren und 39 weitere als frei. Die Belegtheitswahrscheinlichkeit und die Zuverlässigkeit der Zelle werden mittels einer Fuzzy-Bewertung ermittelt. Hierbei wird die Nachbarschaftsbewertung als eine Art Glättungsfilter mit einbezogen. Die Nachbarschaftsberechnung mit Radius eins, wie sie in dieser Arbeit zum Einsatz kommt, bedarf des Zugriffs auf die Umliegenden 27 Zellen des Umweltmodells. Somit müssen insgesamt $(1 + 39) * 27 = 1080$ Zellzugriffe im Octree stattfinden. Trotz einiger Optimierungen dieser Zugriffe durch Speicherung der zuletzt im Octree besuchten Zelle (Bär, 2005), sprengt dies den zulässigen Rahmen, wenn man bedenkt, dass der Laserscanner alle 100 ms 683 neue Abstandswerte liefern kann. Bei Ausnutzung aller möglichen Daten würden pro Millisekunde alleine 7376 Octree-Zugriffe anfallen. Um den Berechnungsaufwand in Grenzen zu halten, wurden bei

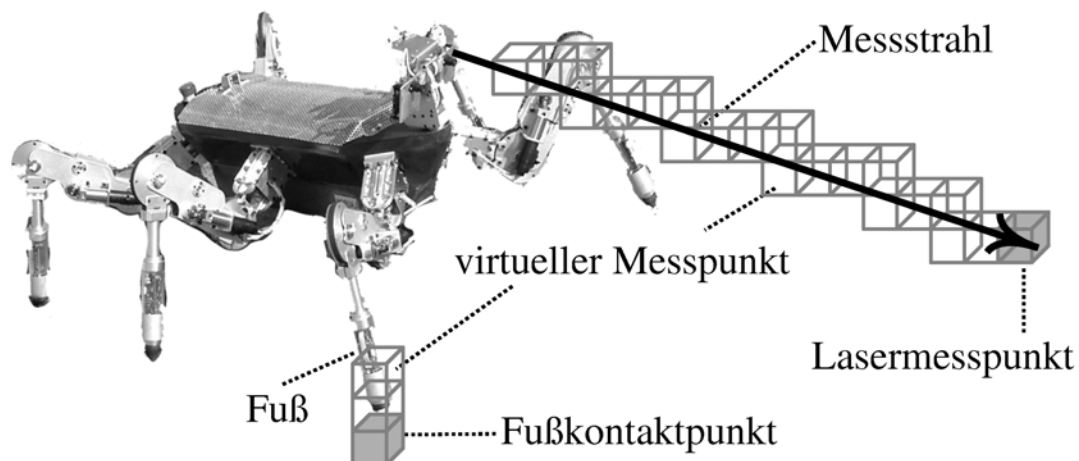


Abbildung 7.12: Durch die realen Sensormessungen (graue Zellen) werden weitere Zellen implizit als frei deklariert (transparente Zellen). Diese werden durch den so genannten „virtuellen“ Sensor mit ins Umweltmodell eingetragen.

der Umsetzung deshalb folgende Optimierungen durchgeführt:

- Der 2D-Lasersensor wird nunmehr erst gezielt platziert und anschließend die Messdatenzeile ins Modell eingetragen.
- Da der Laserscanner einen Öffnungswinkel von 240° aufweist, für das Laufen aber in erster Linie die direkt vor dem Roboter liegenden Zellen von Interesse sind, werden rund 40% der Zeile am Rand vernachlässigt. Es werden nur die Messdaten innerhalb eines Öffnungswinkels von 144° berücksichtigt.
- Die Nachbarschaftsbeziehung beim virtuellen Sensor wird nur auf den letzten 10 cm zum Messpunkt hin berechnet.

Durch diese Maßnahmen wird beim Abscannen der Umgebung etwa alle 1–2 Sekunden eine Messzeile ins Umweltmodell eingetragen.

7.4.3 Aufbau der Karten

Das realisierte Umweltmodell wird nun anhand eines Kartierungsexperimentes evaluiert. Mit diesem Experiment in strukturierter Umgebung soll nachgewiesen werden, dass die Repräsentation der Umwelt, wie sie bei der Aufnahme gewonnen wird, die reale Welt geeignet abbildet und für das Laufen relevante Merkmale extrahierbar sind. Weiterhin soll unter Beweis gestellt werden, dass das erweiterte Inferenzgitter (vergleiche Abschnitt 4.7) sowie die globale $2\frac{1}{2}$ D-Karte (vergleiche Abschnitt 4.8) zu brauchbaren Ergebnissen führen.

Der Versuchsaufbau zu diesem Experiment skizziert Abbildung 7.13. Zu Beginn musste durch Einsatz des Laserscanners die Hindernislandschaft eingescannt werden.



Abbildung 7.13: Der Aufbau des Kartierungsversuches: In einem geraden Gang mit ebenem Boden wurden einige Hindernisse aufgebaut. Eine aus zwei Teilen bestehende Rampe führt auf ein Plateau. An dessen rechter Seite befindet sich eine weitere Erhöhung. Im Anschluss daran ist ein kleiner Graben modelliert. Auf der gegenüberliegenden Seite befindet sich wiederum ein Plateau, ebenfalls mit einer Erhöhung versehen. Abschließend führt eine kurze Rampe wieder vom Plateau auf den Fußboden zurück. Auf der rechten Gangseite ist der Weg durch ein größeres steinartiges Hindernis versperrt. Ganz am Ende des Gangs liegt ebenfalls noch ein steinartiges Hindernis, auf dem sich LAURON im Bild nach getaner Arbeit gerade ausruht.

Dann wurde LAURON über die Hindernisse bewegt. Dabei wurden neben den Laserscannerdaten auch die taktilen Informationen der Füße genutzt, um Daten in die Urkarte einzutragen. Gleichzeitig zur Urkarte wurde die $2\frac{1}{2}D$ -Karte ständig fortgeschrieben.

Die Abbildungen 7.14 und 7.15 zeigen den Aufbau des 3D-Modells anhand von Bilderserien. In Abbildung 7.16 ist das Ergebnis der Kartierung anhand der aufgebauten $2\frac{1}{2}D$ -Karte dargestellt. In der Höhenkarte sind alle Details der Umwelt verzeichnet. Die am tiefsten liegenden Zellen entsprechen dem Fußboden, die höherliegenden Zellen den Plateaus bzw. der ersten Hindernisebene. Die am höchsten liegenden Zellen repräsentieren die seitlichen Wände und die hohen Hindernisse. In den mit Hilfe der Delaunay-Triangulierung erzeugten Ansichten, den unteren Karten mit Dreiecksvermaschung, sind die Höhendifferenzen gut zu erkennen (Abbildungen 7.16(c) und 7.16(d)). In der folgenden Aufzählung wird nochmals von unten beginnend auf einzelne hervorzuhobende Kartendetails eingegangen.

- Die beiden Linien am unteren Kartenrand stellen die nur durch die Fußauftrittstellen bekannte Startposition des Roboters auf dem Fußboden dar.
- Die geteilte Rampe zu Beginn wechselt ihre Farbe gegen Ende, wenn sie die Höhe des Farbwechsels erreicht. Der Boden seitlich davon liegt deutlich tiefer.

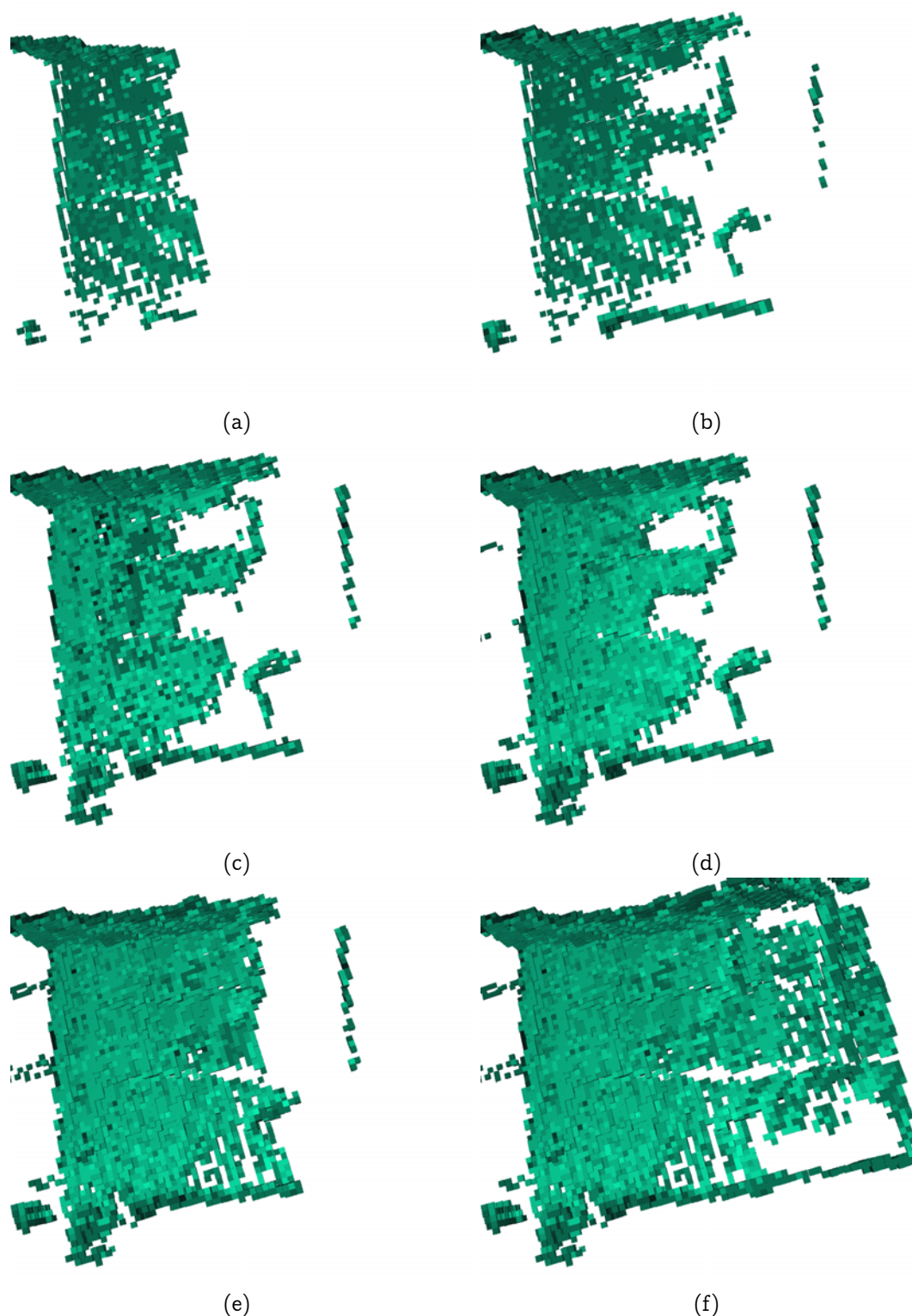


Abbildung 7.14: Kartierungsexperiment: Der schrittweise Aufbau der dreidimensionalen Urkarte wird anhand einer Bilderserie dargestellt. Die Bilder wurden direkt durch die 3D-Visualisierung an der Benutzerschnittstelle erzeugt. Die Helligkeit der einzelnen Zellen entspricht der zugeordneten Belegtheitswahrscheinlichkeit. (a)(b) Das erste Hochschwenken des Lasers im Stand des Roboters. (c)(d) Die Daten werden nach mehrfachem Überstreichen der Zellen sowohl dichter als auch zuverlässiger. (e)(f) Der weitere Aufbau der Karte nachdem der Roboter begonnen hat, in die Szene hineinzulaufen.

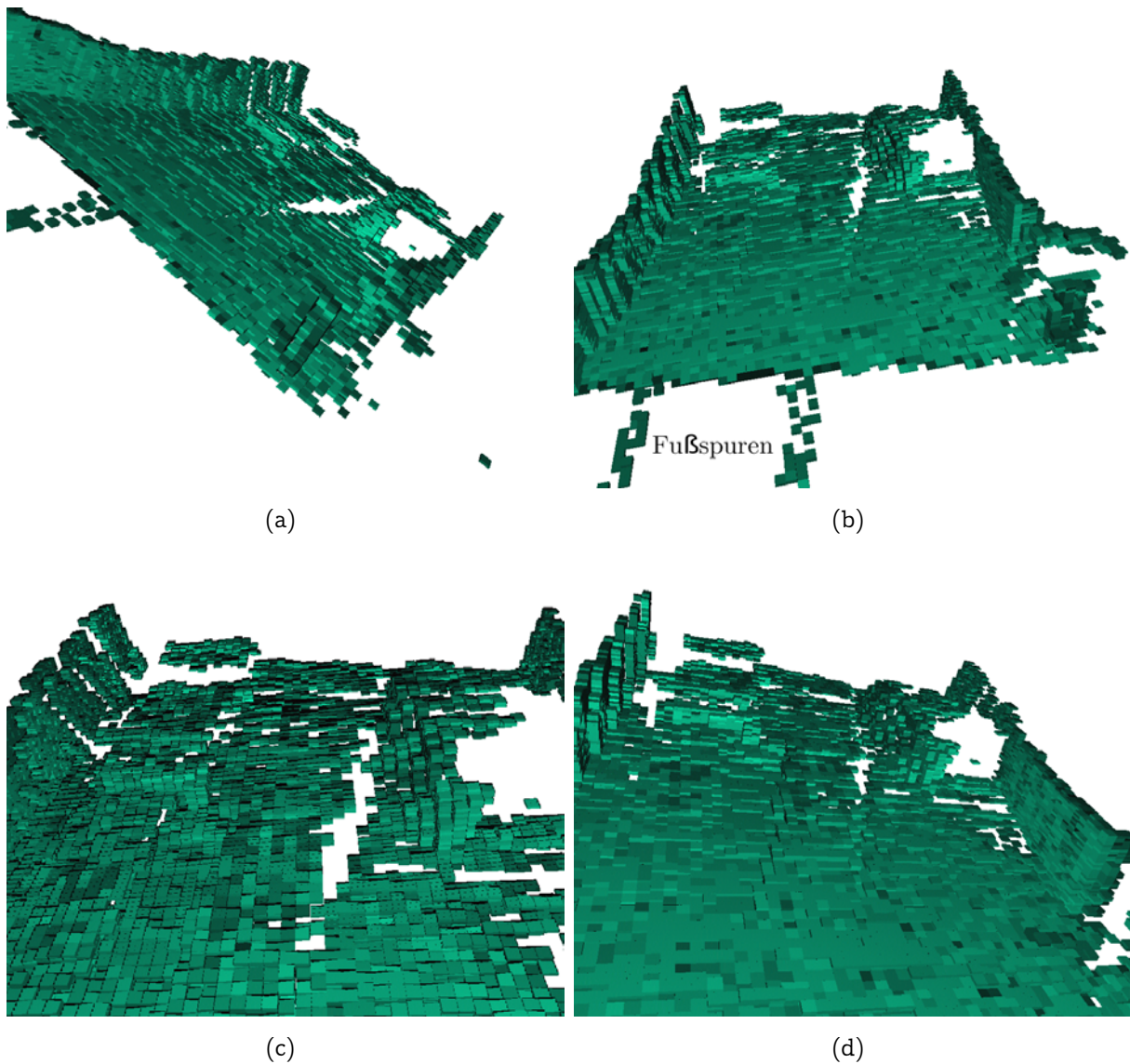


Abbildung 7.15: Kartierungsexperiment: Weitere Ansichten während des Aufbaus des 3D-Modells. Durch rotieren der Ansicht im 3D-Betrachter der Benutzerschnittstelle kann das dreidimensionale Modell inspiziert werden. Die leichten Erhebungen der überschrittenen Hindernisse sind ansatzweise zu erkennen. Die Erhebung des großen Steins neben den Hindernissen auf der rechten Seite tritt hingegen deutlich hervor. Da die Messstrahlen von der Seite auftreffen, liegt von dessen Front keine Information vor. Die beiden Spuren zu Beginn der Karte (b) sind ausschließlich durch Eintragen der Fußaufsetzpunkte entstanden, da an der Startposition keine Laserscanneraufzeichnungen möglich waren.

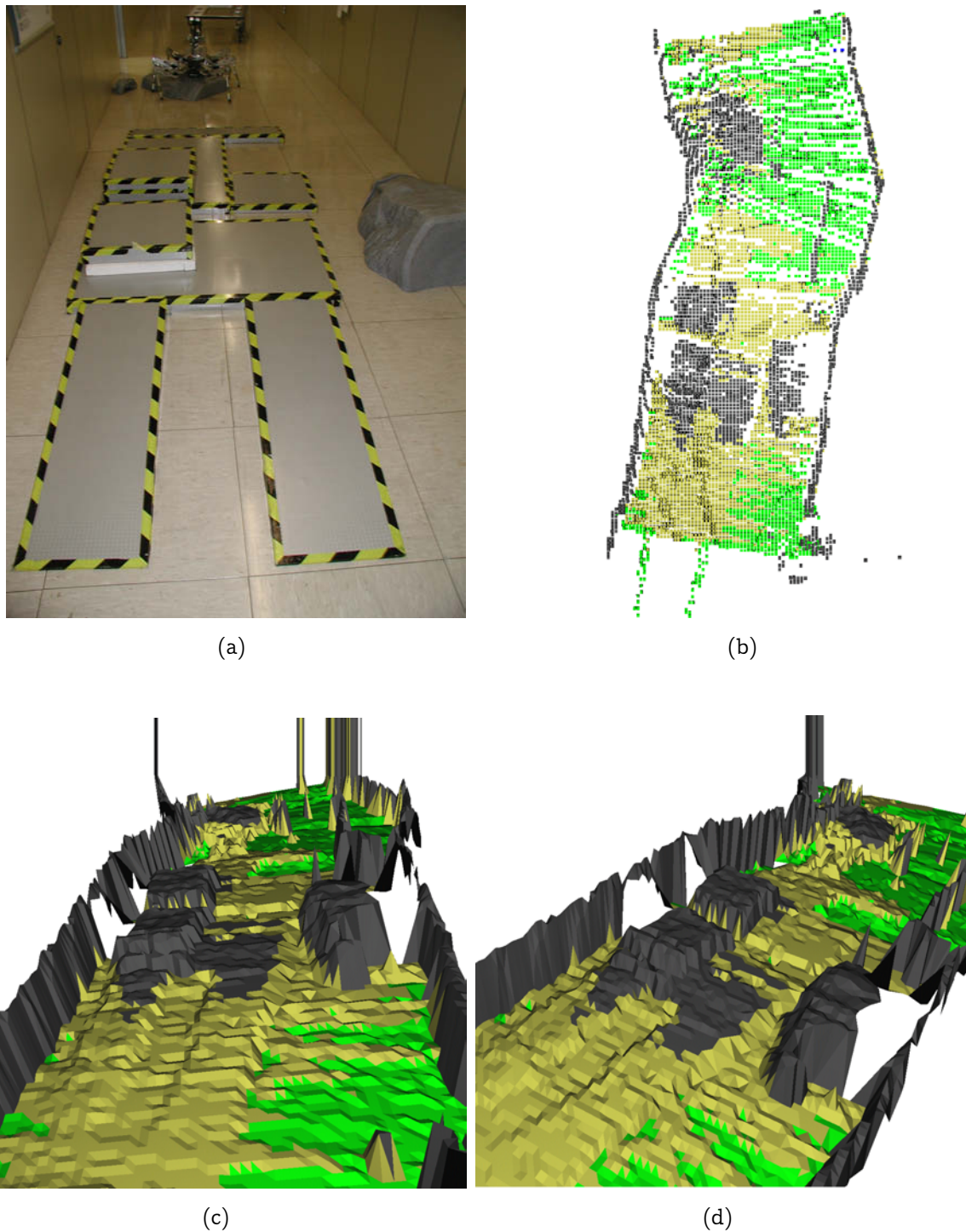


Abbildung 7.16: Das Ergebnis des Kartierungsversuches anhand der generierten globalen $2\frac{1}{2}$ D-Karte: (a) Aufbau des Kartierungsversuches (b) Draufsicht auf die $2\frac{1}{2}$ D-Karte (c) flache Ansicht der $2\frac{1}{2}$ D-Karte (d) $2\frac{1}{2}$ D-Karte von schräg oben. Die Bilder wurden durch verschiedene $2\frac{1}{2}$ D-Kartenansichten der Benutzerschnittstelle erzeugt. Die $2\frac{1}{2}$ D-Karten werden dort dreidimensional dargestellt. Die Farbe des Kubus codiert die in der Höhenkarte eingetragene Höhe.

- Die abrupte Erhöhung der Hindernisse auf dem ersten Plateau ist an der weißen Grenzlinie zu erkennen. Dies gilt in gleichem Maße für den Beginn des Grabens. Die Bodenzellen im Graben selbst verschieben sich durch die perspektivische Projektion unter die höherliegenden Zellen des Grabenendes. Links seitlich von den den Graben bildenden Hindernissen ist keine Umweltinformation vorhanden. Der Boden wird durch die Hindernisse selbst abgeschattet.
- Der große Stein am rechten Rand ist deutlich hervorgehoben. Dessen schräge nach außen abfallende Oberfläche liegt ebenfalls im Schatten des Lasers.
- Beim überschreiten der Hindernisse kann LAURON verrutschen. Die Verzerrungen der Karte auf Höhe der Hindernisse nach dem Graben lassen sich hierdurch erklären.
- Am Abgang vom zweiten Plateau kommt es zu starken Verzerrungen der Karte, als ob der Gang hier eine Kurve hätte. Dem ist aber nicht so, der Gang ist völlig gerade. Da an der seitlichen Wand an dieser Stelle der Sicherungskasten angebracht ist und unter den Bodenplatten sowie in der Decke Strom-, Netzkabel und Eisenrahmen den Weg kreuzen, ist eine externe Störung des Magnetfeldes wahrscheinlich.

7.4.4 Zusammenfassung und Erkenntnisse

Das entworfene Umweltmodell hat sich im konkreten Einsatz mit LAURON IV bewährt. Die durchlaufene Umgebung wird in einer lokalen dreidimensionalen Urkarte in guter Näherung abgebildet. Das entwickelte erweiterte Inferenzgitter eignet sich zur Fusion unterschiedlicher Sensordaten. Die schritthaltende Aktualisierung der globalen $2\frac{1}{2}$ D-Karte erlaubt der Planungsebene den Zugriff auf aktuelle Daten.

Die Korrektheit der globalen Karte leidet jedoch stark unter dem hohen Rauschanteil der Sensordaten zur Erfassung der Roboterorientierung. Die lokale Planung des Roboters, die Adaption, die für den direkt vor dem Roboter liegenden Kartenbereich zuständig ist, ist hiervon nicht betroffen, da die direkt vor dem Roboter liegenden Bereiche durch zyklische Messungen ständig aktuell gehalten werden. Um eine Bahnplanung auf fehlerhaften Umweltdaten zu vermeiden, muss die Bahnplanung unterwegs neu durchgeführt werden, um den geplanten Weg mit aktuellen Daten zu verifizieren.

7.5 Bahnplanung

Mit der im letzten Abschnitt erzeugten $2\frac{1}{2}$ D-Karte kann der in Kapitel 5 entwickelte Bahnplaner einen Weg vom Startpunkt hin zu einem vorgegebenen Ziel planen. Um die Leistungsfähigkeit dieses Bahnplaners zu untersuchen, wird zunächst nicht auf reale Umweltdaten zurückgegriffen werden. Die Experimente beruhen auf manuell erstellten exakten Karten, um von Kartierungsungenauigkeiten wie sie im letzten Abschnitt zu beobachten waren auszuschließen. Es stehen vielmehr die Algorithmen zur Bahnplanung

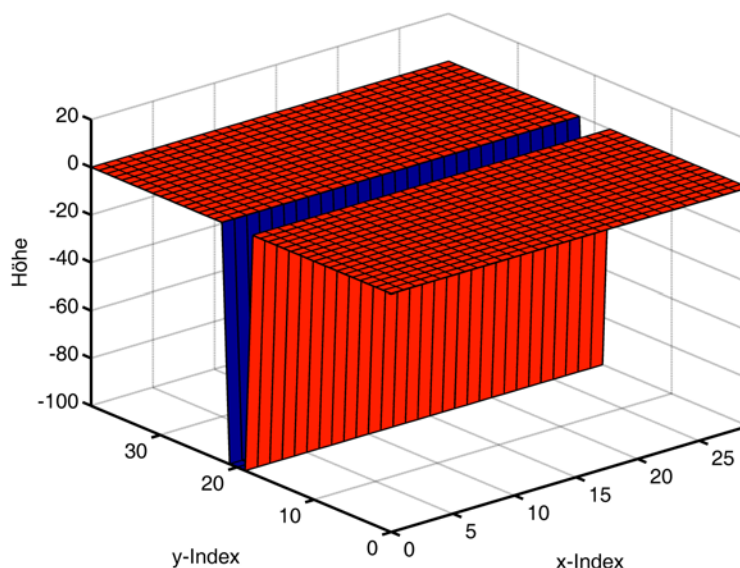


Abbildung 7.17: Verlauf des Testgrabens mit einer Breite von 3 Zellen.

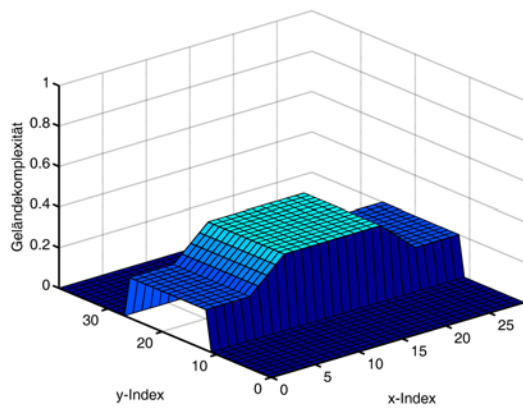
im Mittelpunkt, insbesondere die Bewertungsfunktion zur Ermittlung der Geländekomplexität.

Um die dargestellten Experimente korrekt zu interpretieren, sei darauf hingewiesen, dass die Bewertung für eine bestimmte Roboterposition \mathbf{p} , gekennzeichnet durch den x - und y -Index der Zelle c des Körpermittelpunkts und eine der beiden möglichen Orientierungen Nord oder Ost, stets in eine Mittelpunktzelle eingetragen wird. Hierbei steht der Begriff Komplexität für die in Abschnitt 5.6.4 definierte 3D-Geländekomplexität. Die abgebildeten Komplexitätskarten entsprechen schließlich dem zur Bahnplanung genutzten abstoßenden Potenzial.

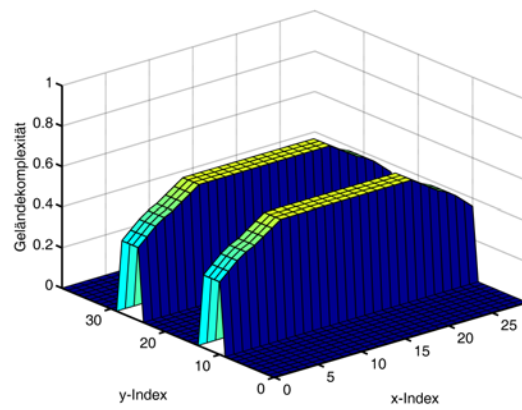
7.5.1 Graben

Als Beispiel soll das Überschreiten eines Grabens untersucht werden. Hierbei zeigt sich, inwiefern sich die Breite des Grabens auf die Bewertung der Geländekomplexität auswirkt. Der Graben hat eine Tiefe von 1 m und liegt quer zum Roboter, falls dieser die Orientierung Nord annimmt (Abbildung 7.17). Die Vertiefung beginnt in y -Richtung bei Index 19.

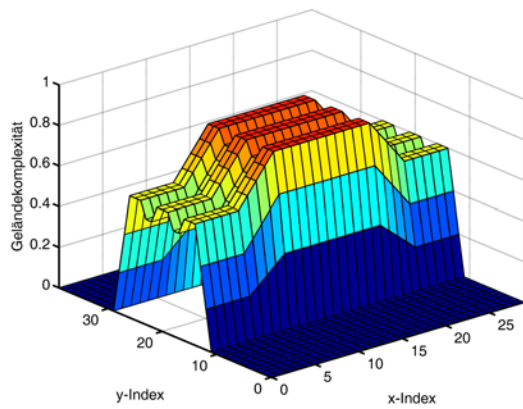
Abbildung 7.18 zeigt die resultierenden Bewertungen der Geländekomplexität. Diese steigt an, sobald Arbeitsbereiche von Beinen sich mit Zellen des Grabens überschneiden. Bei einer Grabenbreite von einer Zelle bei der Orientierung Nord (Abbildung 7.18(a)) bildet sich ein Plateau aus: jeweils zwei gegenüberliegende Bein-arbeitsbereiche treffen auf den Graben. Wird der Graben auf 3 Zellen ausgeweitet (Abbildung 7.18(c)), steigt die Komplexität an, da weniger gültige Konfigurationen möglich sind. Weil an einigen Stellen vier Arbeitsbereiche gleichzeitig den Graben berühren, bilden sich drei kammartige Maximas aus. Bei einer Grabenbreite von 6 Zellen (Abbildung 7.18(c)) kommen ganze Bein-arbeitsbereiche im Graben zum liegen, so dass keine gültige Konfiguration



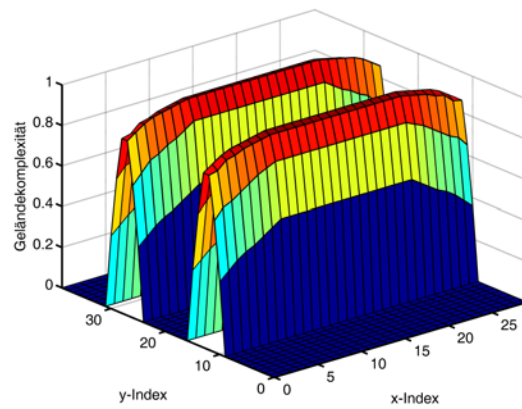
(a)



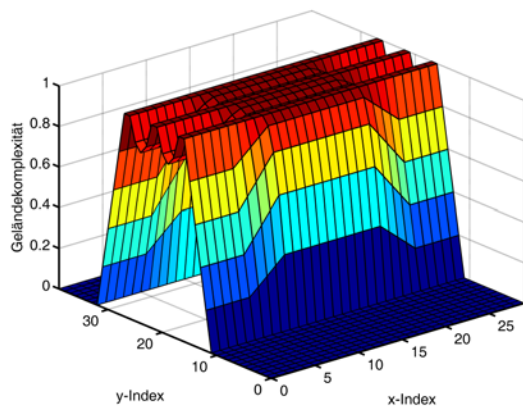
(b)



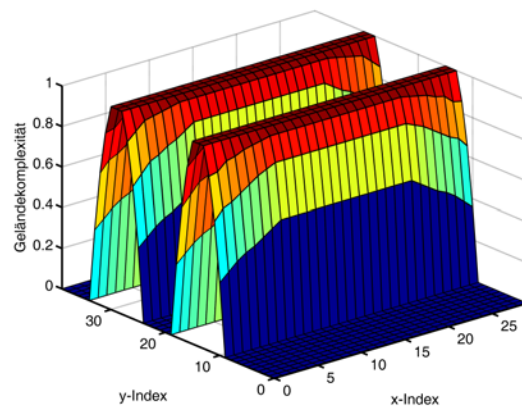
(c)



(d)



(e)



(f)

Abbildung 7.18: Bewertung der Grabenexperimente mit einer Grabenbreite von (a)(b) einer Zelle (c)(d) 3 Zellen (e)(f) 6 Zellen. Aufgetragen ist jeweils die Geländekomplexität für die Orientierungen (a)(c)(e) Nord und (b)(d)(f) Ost.

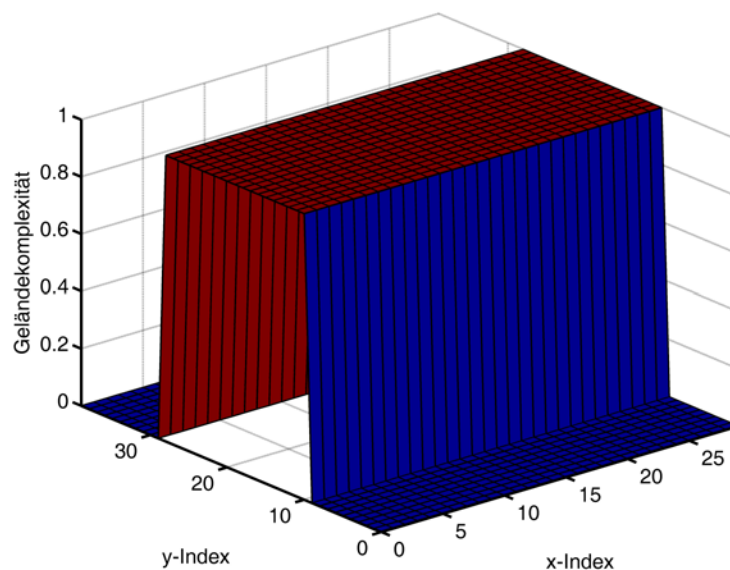


Abbildung 7.19: Bewertung der Grabenexperimente nach Bai et al. (2002): Aufgetragen ist die Geländekomplexität für eine Grabenbreite von einer Zelle.

möglich ist. Die Geländekomplexität steigt auf den Maximalwert von 1.

Bei der Orientierung Ost (Abbildungen 7.18(b), 7.18(d) und 7.18(f)) lässt sich Ähnliches beobachten. Allerdings treten die hohen Komplexitätswerte parallel entlang des Grabens auf, wenn die drei Arbeitsbereiche einer Körperseite den Graben berühren. Weil mehr Zellen des Grabens die Arbeitsbereiche schneiden, liegen die Komplexitäten höher als bei der Orientierung Nord.

Zu Vergleichs- und Testzwecken ist auch das Bahnplanungsverfahren nach Bai et al. (2002) implementiert worden. Schon bei einer Grabenbreite von einer Zelle (Abbildung 7.19) wird der Graben auf Grund der Definition der positiven Hindernisse als nicht überschreitbar bewertet.

7.5.2 Säule

An einem weiteren Beispiel soll die Problematik von unüberwindbaren Hindernissen geringer Ausdehnung wie beispielsweise ein dünner Baum untersucht werden. Hierzu wird eine quadratische Säule mit einer Höhe von 50 cm und einer Grundfläche von vier Zellen modelliert (Abbildung 7.20).

Unter Berücksichtigung der Definition der Einflussgrenzen, umgibt die Säule ein Bereich mit maximaler Geländekomplexität (Abbildung 7.21). Die Laufmaschine meidet dieses Hindernis. Ohne die Berechnung der Einflussgrenze, wird lediglich der Bereich des Zentralkörpers mit maximaler Komplexität bewertet, da der Körper nicht über die Säule hinweg bewegt werden kann (Abbildung 7.22). Die hohen Bewertungen, die an der Fläche des Zentralkörpers auf der Höhe der Schulterzellen angrenzen, entstehen durch das Beinmodell. Die Säule bildet hier die Auflagezellen eines Beines.

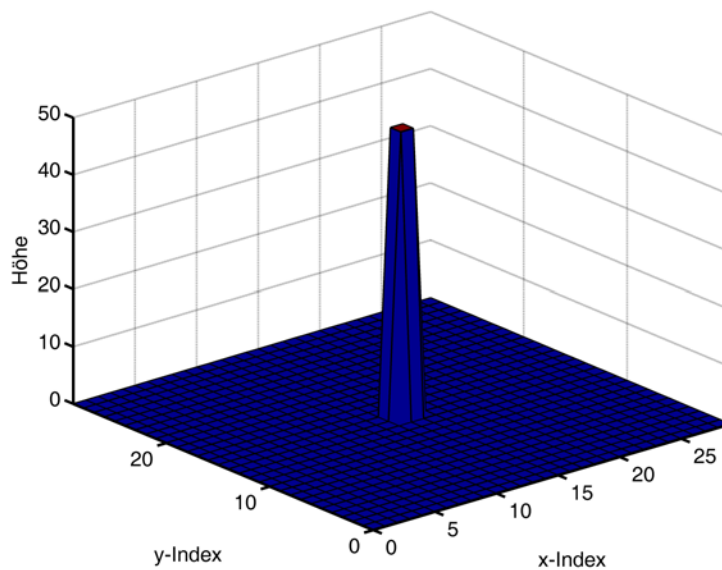


Abbildung 7.20: Aufbau des Säulenexperiments. Eine 50 cm hohe Säule inmitten einer Ebene. Diese Säule ist quadratisch und umfasst vier Zellen.

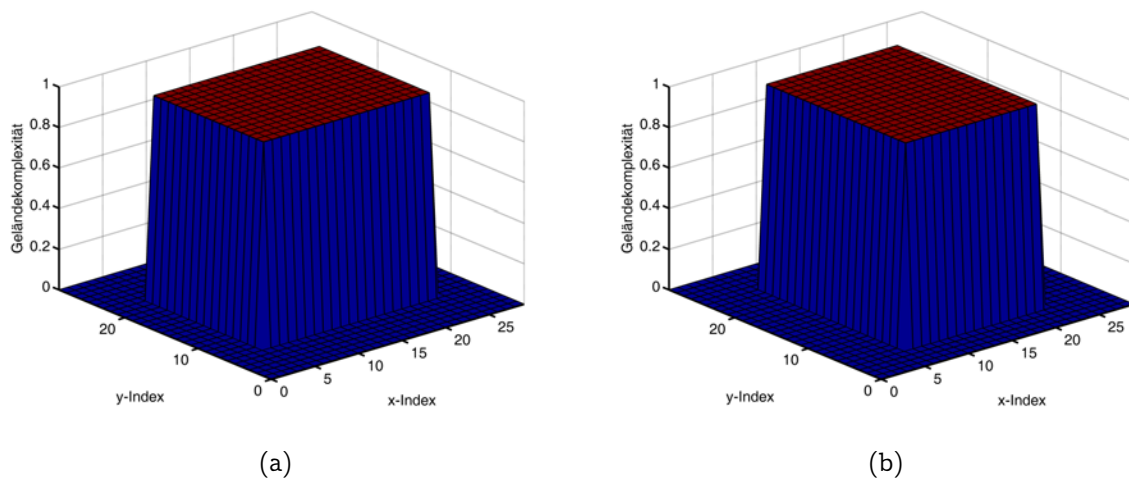


Abbildung 7.21: Säulenexperiment mit Einflussgrenze: Aufgetragen ist die Geländekomplexität für die Orientierungen (a) Nord und (b) Ost. Aufgrund der Einflussgrenzen entsteht um die Säule ein Gebiet maximaler Geländekomplexität.

7.5.3 Bahnplanung

Nachdem alle bisherigen Experimente im Wesentlichen bestimmte Eigenschaften und Eigenheiten der Komplexitätsbewertung zum Gegenstand hatten, soll nun der Fokus auf die Nutzung der Bewertung zur Bahnplanung gerichtet werden. Die hierfür modellierte Karte (Abbildung 7.23) enthält eine Treppe mit einer Stufenhöhe von 8 cm im oberen Teil, einen Graben der Breite einer Zelle in der Mitte, eine Säule wie im letzten Beispiel sowie ein Hindernis der Höhe 25 cm, das von Zellen der Höhe 10 cm umgeben ist, um die Ausbildung von Einflussgrenzen zu vermeiden.

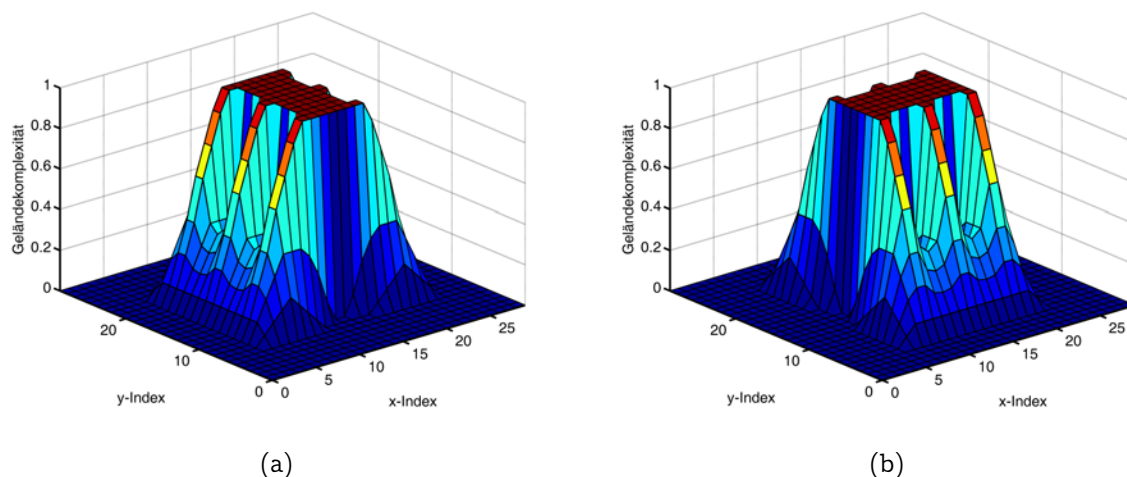


Abbildung 7.22: Säulenexperiment ohne Einflussgrenze: Aufgetragen ist die Geländekomplexität für die Orientierungen (a) Nord und (b) Ost. Die Region um die Säule wird weniger restriktiv bewertet. Der Zentralkörper des Laufmaschinenmodells zeichnet sich deutlich ab.

Die berechneten Geländekomplexitäten sind in den Abbildungen 7.24(b) und 7.24(c) dargestellt. Das Gebiet um die Säule und die Gebiete neben der Treppe sind auf Grund der Einflussgrenzen nicht betretbar. Die Treppe selbst erhält niedrigere Bewertungen, wenn sich die Laufmaschine in der Nähe der Stufenmitten aufhält und dabei die Orientierung Nord einnimmt. Bei anderer Orientierung wird die gesamte Treppe sehr schlecht bewertet, d.h. ein Besteigen der Treppe seitwärts ist nicht möglich.

Auffällig ist ein Block von Einer-Komplexitäten am linken, unteren Rand bei Orientierung Nord. Verantwortlich dafür ist das 25 cm hohe Hindernis. Befindet sich die Laufmaschine zu weit links, stehen die linken Beine auf der Ebene, während mindestens ein rechtes Bein das hohe Hindernis in seinem Beinarbeitsbereich hat. Die Folge ist das Ausbilden einer Einflussgrenze. Dieses Hindernis kann aber dennoch überwunden werden, sofern sich die Laufmaschine mehr rechts positioniert. Das spiegelt sich in den niederen Potenzialen wider.

7.5.3.1 Vollständig bekannte Karte

Die modellierte Karte sei der Laufmaschine vollständig bekannt. Als Start dient die Zelle (1, 1); das Ziel markiert die Zelle (20, 32) in der Mitte der Treppe. Neben dem abstoßenden Potenzial gemäß Abbildungen 7.24(b) und 7.24(c), welches mit einem Faktor $K_r = 10$ gewichtet wird, ist auch das anziehende Potenzial des Ziels aktiv. Dieses wird ebenfalls mit dem Faktor $K_a = 10$ verstärkt. Dadurch sind beide Potenziale gleichberechtigt. Unter dem Einfluss der sich überlagernden Potenzialfelder und einer maximal erlaubten Geländekomplexität $A_{\max} = 0,5$, ergibt sich der schwarz eingefärbte Weg in Abbildung 7.24(a). Der Roboter läuft zunächst mit der Orientierung Nord am größeren Hindernis im unteren Kartendrittel vorbei. Danach dreht sich der Roboter und läuft mit Orientierung Ost am Rand des Grabens entlang, bis er diesen in der Mitte überquert und schließlich die Treppe mit der Orientierung Nord Stufe für Stufe besteigt.

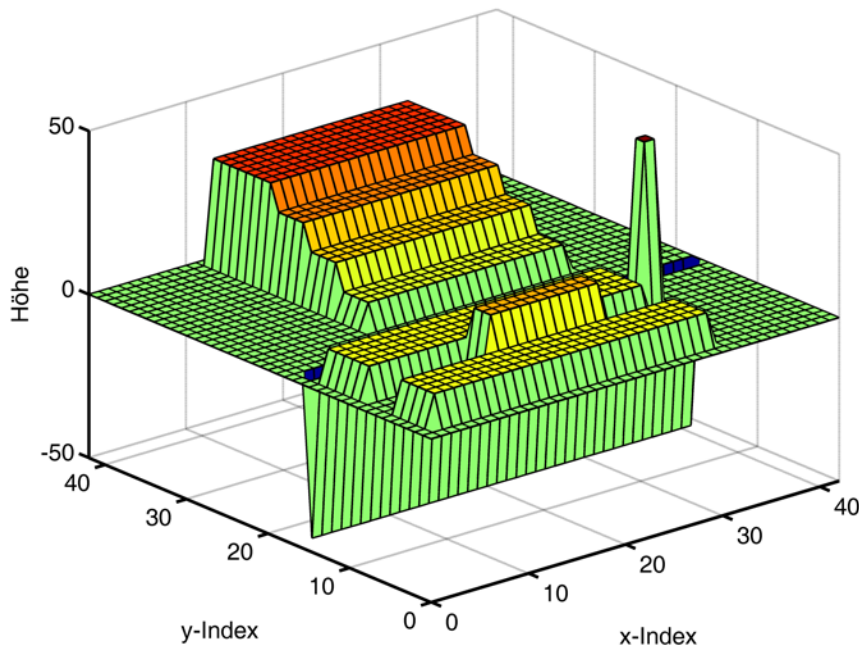


Abbildung 7.23: Aufbau des Bahnplanungsexperiments: Die Karte umfasst eine Treppe (oben), einen schmalen Graben (Mitte), eine Säule (rechts) und Hindernisse unterschiedlicher Höhe (links).

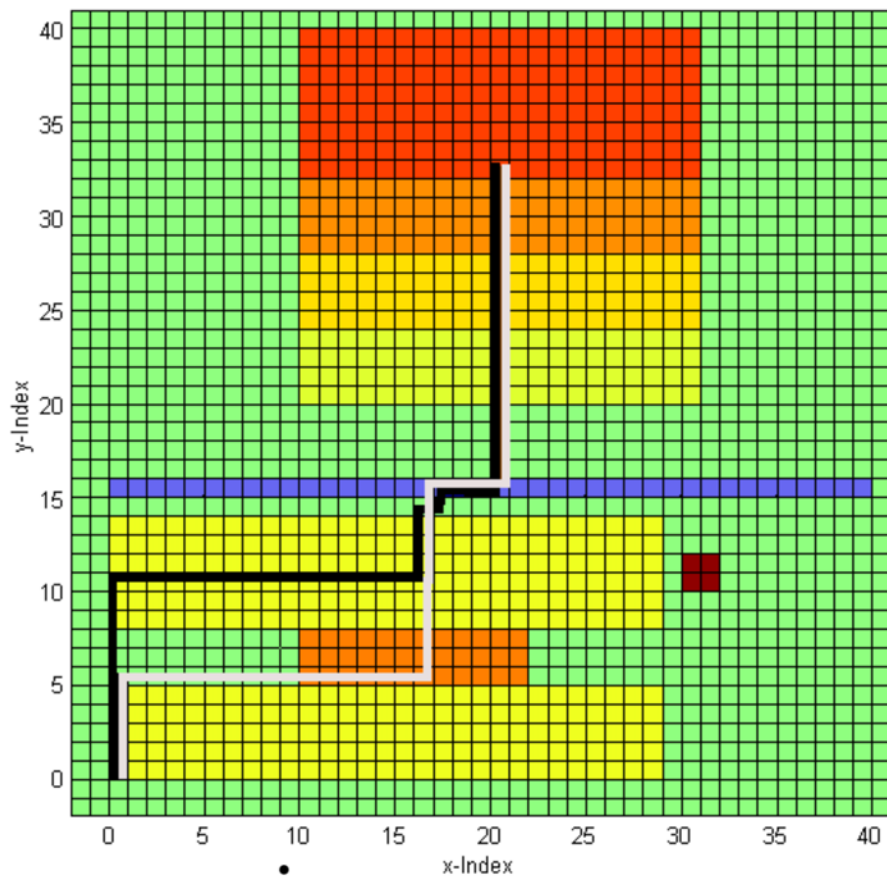
Wird der Einfluss des anziehenden Potentials verstärkt ($K_a = 40$), wählt der Roboter einen kürzeren Weg zum Ziel. Das anfängliche Hindernis wird nun überstiegen, was durch den grauen Weg in Abbildung 7.24(a) dargestellt ist.

7.5.3.2 Unvollständige Karte

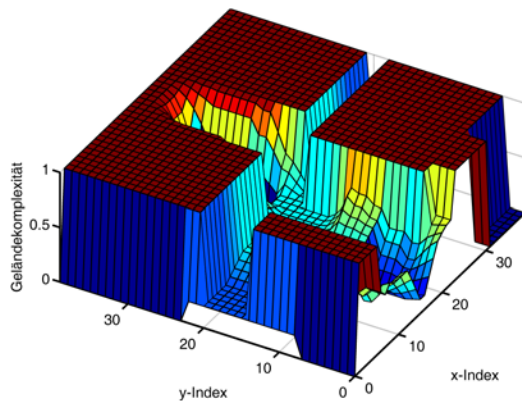
Um die Fähigkeit des vorliegenden Bahnplaners zu demonstrieren, auch unbekannte Bereiche einer Karte bei der Wegsuche zu berücksichtigen, wird eine rechteckige Fläche der Karte als unbekannt gekennzeichnet. Dieses Gebiet und das daran angrenzende erhält als Komplexität den Explorationswert $e = 0,5$ (Abbildungen 7.25(b) und 7.25(c)). Da die maximal erlaubte Geländekomplexität A_{\max} diesen Wert nicht unterschreitet, kommt unbekanntes Gelände für die Wegsuche in Betracht. Der geplante Weg, der direkt durch das unbekannte Terrain verläuft, ist in Abbildung 7.25(a) skizziert.

7.5.4 Zusammenfassung und Erkenntnisse

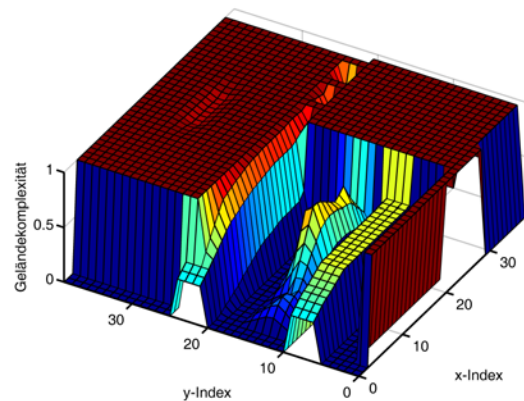
Die Eignung des konzipierten Bahnplaners wurde anhand mehrerer Experimente bestätigt. Zahlreiche Hindernissituationen wurden entsprechend den Fähigkeiten und Eigenschaften einer Laufmaschine korrekt beurteilt. Die Berücksichtigung der Wechselwirkung aller Beine durch die Bewertungsfunktion ermöglicht das Erkennen von besteigbaren Hindernissen und überschreitbarem Gelände. Mit der Einbeziehung der Körperneigung kann die Anzahl der unzugänglichen Objekte weiter reduziert werden. Gerade



(a)

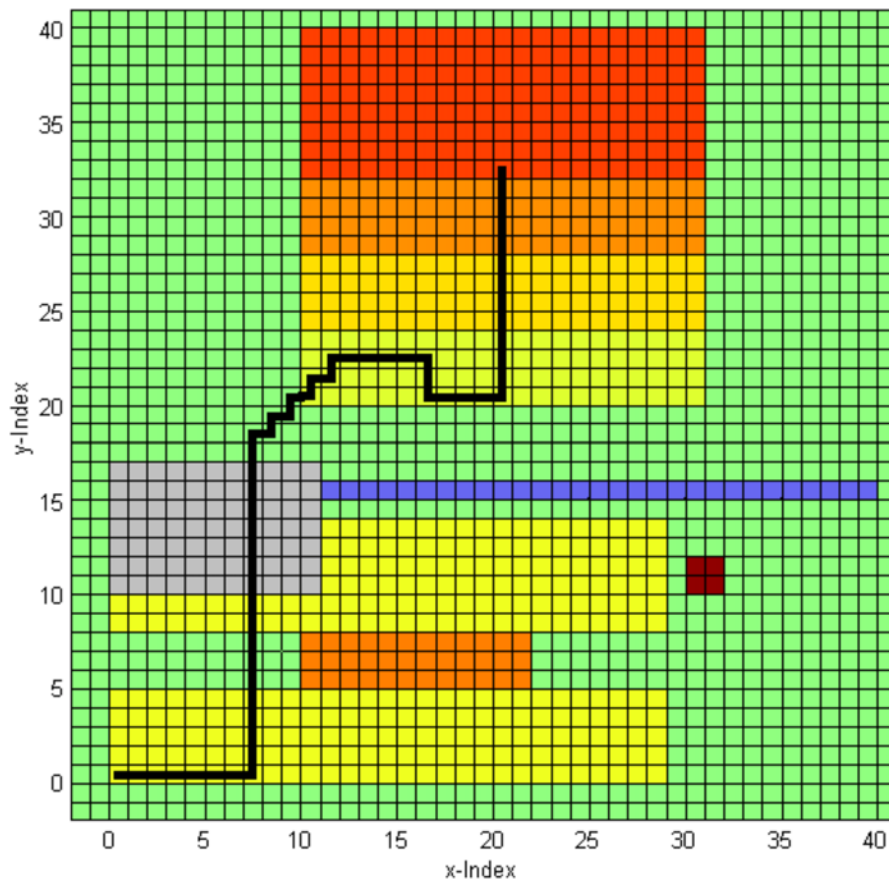


(b)

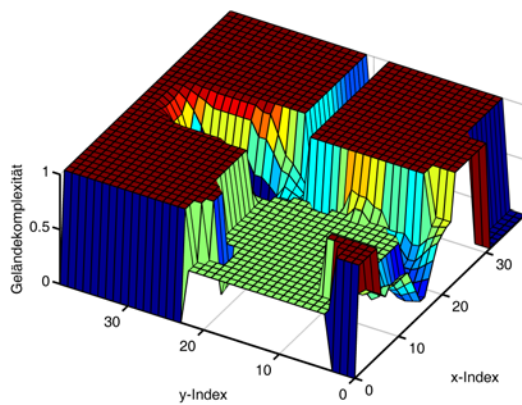


(c)

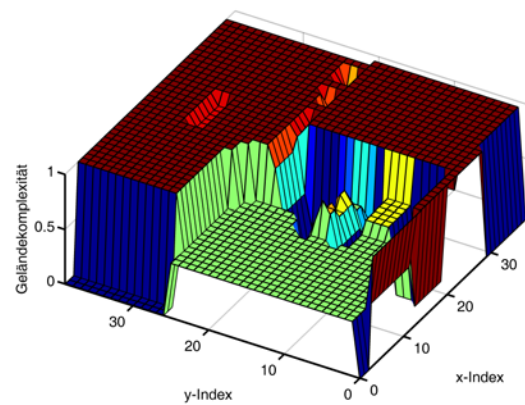
Abbildung 7.24: Bahnplanung bei vollständig bekanntem Umweltmodell. (a) Die geplanten Wege bei vollständig bekanntem Umweltmodell. Schwarzer Weg: Die Faktoren für anziehendes und abstoßendes Potenzial sind gleich. Grauer Weg: Der Faktor für das anziehende Potenzial ist viermal größer als der Faktor des abstoßenden Potenzials. (b)(c) Komplexität der Gesamtkarte aus Abbildung 7.23: Aufgetragen ist die Geländekomplexität für die Orientierungen (b) Nord und (c) Ost.



(a)



(b)



(c)

Abbildung 7.25: Bahnplanung bei unvollständigem Umweltmodell. (a) Das graue Rechteck markiert ein unbekanntes Gebiet. Der geplante Weg führt hindurch. (b)(c) Der unbekannte Bereich wird bei der Bewertung ausgedehnt. Seine Ausdehnung ist durch die gebildete Fläche erkennbar. Aufgetragen ist die Geländekomplexität für die Orientierungen (b) Nord und (c) Ost.

Treppen, welche hier stellvertretend für alle Schrägen stehen, lassen sich dadurch bis zu einer gewissen Steigung erklimmen. Weiterhin gelang das Anpassen des Konzepts der Einflussgrenzen für den neuen Bewertungsablauf. Die Fähigkeit, auch ungesehene Bereiche einer Karte durch den Explorationswert in die Planung mit einzubeziehen, gestattet Wege jenseits bekannter Gebiete. Das Fehlen der globalen Bahnplanung lässt sich so kompensieren.

7.6 Adaption an das Gelände

Die in Kapitel 6 vorgestellten Methoden zur Anpassung des Laufverhaltens an die Geländebedingungen sollen an dieser Stelle einer Verifikation unterzogen werden. Hierzu wurde sowohl die Geländebeurteilung während des in Abschnitt 7.4 kartierten Hindernisparcours als auch die durch die Adaption vorgeschlagenen Laufparameter mitprotokolliert.

7.6.1 Geländebewertung

Die Eingangsvariablen für die Fuzzy-Bewertung der Laufparameter entsprechen den aus der Höhenkarte abgeleiteten Geländebewertungen und der Selbstreflexion der reaktiven Steuerung (vergleiche Abschnitt 6.3). Das für die Adaption zu bewertende Gebiet wurde an die Parameter von LAURON angepasst und umfasst einen Bereich von einem Quadratmeter um den Robotermitelpunkt. Somit wird gewährleistet, dass beispielsweise eine direkt neben dem Roboter befindliche Wand die Bewertung nicht verfälscht. Es wird nur das Gebiet bewertet, das von LAURON auch betreten wurde. Die Bewertung wird nicht ständig durchgeführt, sondern nur dann, wenn sich der Robotermitelpunkt aus dem zuvor bewerteten Gebiet herausbewegt.

Abbildung 7.26 stellt die während des Kartierungsversuches ermittelten Eingangsgrößen für die Fuzzy-Bewertung dar. Die Definition der zu den linguistischen Variablen zugehörigen Fuzzy-Mengen können dem Anhang C entnommen werden.

Erst als LAURON den zuvor im Stand kartierten Bereich betritt und somit die Exploration ein Mindestmaß überschreitet, liefert die Geländebewertung gültige Daten. Nach einigen Schritten steigt die Exploration sogar auf eins: der im Stand ausgiebig kartierte Bereich weist nur noch wenig Lücken auf. Danach schwankt die Exploration zwischen 70% und 95%. Der kurzzeitige Einbruch bei Zeitindex 7000 ist durch den Graben zu erklären: der Boden des Grabens liegt zunächst im Schatten des Laserscanners. Erst beim Überwinden selbst werden einige Lücken im Modell geschlossen. Auffällig ist noch der lange konstante Bereich: das Überwinden der großen Hindernisse in der Mitte des Parcours hat wesentlich mehr Zeit in Anspruch genommen, als die Strecken zu Beginn und am Ende des Laufs. Deshalb wurde über längere Zeit keine erneute Gebietsbewertung durchgeführt.

Die Zuverlässigkeit der Karte bewegt sich immer zwischen 50% und 70%. Durch das Rauschen der Laserabstandswerte treten kleinere Schwankungen in den Höhenwerten auf, so dass die Zuverlässigkeit in der realen Anwendung nie die 100%-Marke erreichen

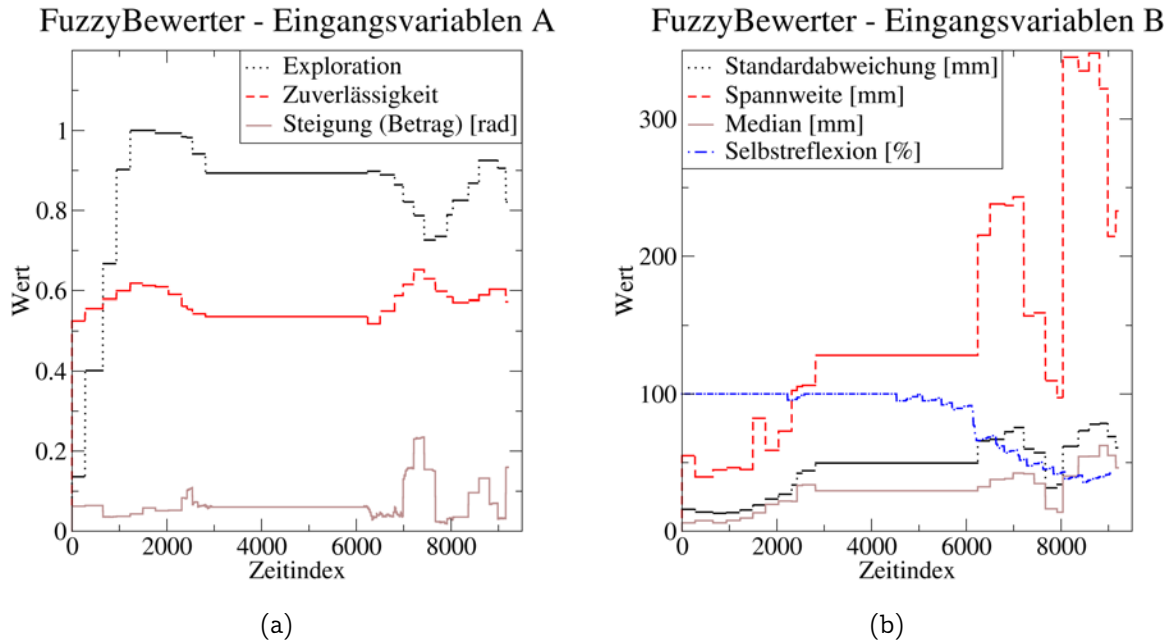


Abbildung 7.26: Eingangsgrößen der Laufparameter-Bewertung: (a) Exploration, Zuverlässigkeit und der Betrag der Steigung (b) Standardabweichung, Spannweite, Median und Selbstreflexion der reaktiven Steuerung

kann.

Bis auf die beiden großen Steigungen bei Zeitindex 7000 (Graben) und Zeitindex 8500 gegen Ende im Bereich des großen Hindernisses, sowie einigen kleinen Ausreißern zu Beginn durch das auf den Graben zuführende Hindernis, bleibt der Steigungsbetrag in etwa bei 4° , was der Unebenheit des gesamten Aufbaus zuzuschreiben ist. Während die kleinste Quadrate Ebene der Gebietsbewertung über längere Zeit hinweg konstant bleibt, wird der Steigungsbetrag bei sich ändernder Roboterorientierung ständig neu berechnet.

Da auf der Teststrecke kaum Ausreißer vorhanden sind, zeigen Standardabweichung, Median und Spannweite einen ähnlichen Verlauf. Bis zum Betreten der großen Hindernisse, am ersten kleinen Einbruch bei der Selbstreflexion durch auftretende kurze Beinkollisionen bei Zeitindex 2000 erkennbar, steigen sie kontinuierlich an. Während LAURON der Graben größere Probleme bereitet und die Selbstreflexion dabei abfällt, bewegt sich der Roboter kaum vorwärts. Es wird keine neue Gebietsbewertung durchgeführt. Erst als sich LAURON bei Zeitindex 6000 direkt über dem Graben befindet, findet eine neue Berechnung statt, was zu einem sprunghaften Anstieg der Bewertung führt. Da die Mittelbeine den Graben noch überschreiten und gleichzeitig die Vorderbeine eine Bodensuche durchführen, fällt die Selbstreflexion immer weiter. Ist die Hindernislandschaft überwunden, erholt sich diese Bewertung nur langsam, um nicht erneut in kritische Situationen zu kommen. An dieser Stelle fallen auch die Geländewertungen wieder ab, bis schließlich mit dem Hindernis am Ende eine große Unstetigkeit in der Umgebung auftaucht.

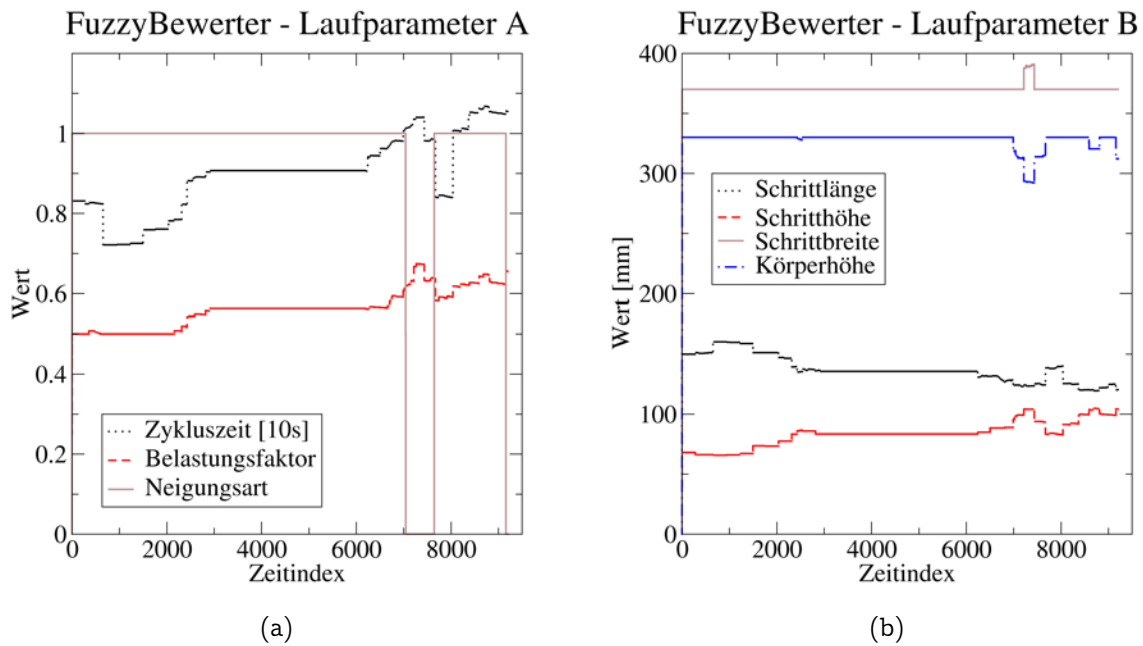


Abbildung 7.27: Ausgangsgrößen der Laufparameter-Bewertung: (a) Zykluszeit, Belastungsfaktor und Neigungsart (b) Schrittlänge, -höhe, -breite und Körperhöhe

7.6.2 Anpassung der Laufparameter

Auf Basis der im letzten Abschnitt diskutierten Geländebewertung werden durch den Fuzzy-Bewerter die auszugebenden Laufparameter bestimmt. Die protokollierten Daten sind Abbildung 7.27 zu entnehmen. Die zu Grunde liegenden Fuzzy-Regeln sowie die definierten Fuzzy-Mengen sind in Anhang C beschrieben. Grundsätzlich wurden sie sehr behutsam gewählt, um den Spielraum der Automatik in Grenzen zu halten.

Die Komplexität des Geländes lässt sich am besten an der Zykluszeit ablesen. Dies ist der geeignetste Parameter zur Geschwindigkeitsanpassung. Zu Beginn darf LAURON schnell laufen. Mit dem Auftreten des ersten echten Hindernisses auf dem Plateau wird die Zykluszeit jedoch schlagartig reduziert. Im weiteren Verlauf beim Übersteigen des Grabens bei Zeitindex 7000 wird auf Grund der gemessenen Steigung auch die Neigungsart geändert: der Roboter richtet sich nunmehr an der geschätzten Bodenneigung aus anstatt waagrecht zur Erde. An dieser Stelle erfährt auch der Belastungsfaktor eine Änderung hin zur Gangart „Tetrapod“. Nach dem Grabenhindernis entspannt sich die Lage bis am Ende das Hindernis wieder ein behutsameres Laufen bedingt. Für die Schrittparameter und die Körperhöhe gilt entsprechendes. Die Möglichkeit zur Anpassung dieser Parameter wurde jedoch deutlich restriktiver gehalten.

7.6.3 Fußaufsetzpunktplanung

Die in Abschnitt 6.5 beschriebene Methode zum dedizierten Setzen von Fußpunkten bedarf zur Umsetzung auf einem realen System noch die Möglichkeit, den Sensoreinsatz gezielt zu steuern. Hierzu benötigt das System eine Instanz zur Sensoreinsatzplanung. Diese bestimmt und steuert die Strategie mit der die Umwelt durch den Laserscanner

erfasst werden soll. Eine Basisstrategie, die beispielsweise zu Beginn des Kartierungsexperimentes zur Anwendung kam, ist das großflächige Abscannen des vorausliegenden Gebietes. Während des Laufens ist eine derartige Strategie eher hinderlich, da ein komplettes Auf- und Abschwanken des Kopfes zu lange dauern würde. Der Blick wäre zu lange von der relevanten Umgebung direkt vor dem Roboter gerichtet. Beim Laufen wird der zu erfassende Bereich daher auf die unmittelbar vorausliegenden 30–40 cm beschränkt.

Findet die Fußaufsetzpunktplanung ein nicht ausreichend exploriertes Gebiet vor, so muss die aktuelle Strategie unterbrochen werden, um zunächst die für das Setzen des nächsten Schrittes unabdingbare Information zu sammeln. Dies erfüllt die in Abschnitt 4.1 formulierte Anforderung nach „gezielter Nachmessmöglichkeit“. Umgesetzt wird dies derart, dass der auf dem Kopf befindliche Sensor die Umgebung der unsicheren Stelle erneut abscannt, wodurch eine Verbesserung der Qualität der Daten zu erwarten ist. Es versteht sich, dass dies derzeit nur für den Fall der beiden Vorderbeine realisiert werden kann – die voraussichtlichen Aufsetzpunkte von Mittel- und Hinterbeinen liegen außerhalb des Sichtbereiches des am Kopf angebrachten Laserscanners.

7.6.4 Zusammenfassung und Erkenntnisse

Die Funktionstüchtigkeit der Gebietsbewertung auf Basis der 2½D-Karte konnte beim Durchqueren eines Hindernisparcours bestätigt werden. Ebenso wurde gezeigt, dass die entwickelte automatische Adaption der Laufparameter nachweislich zu korrekten Ergebnissen führt.

Die vorgestellten Methoden runden die Navigation für Laufmaschinen ab und zeigen, dass die mit Hilfe der Lokalisation und Umwelterkennung gewonnenen Informationen sinnvoll weiterverarbeitet werden können. Wie schon während der Konzeption der Adaption bemerkt, sind die Möglichkeiten der Anpassung von Laufverhalten an die Geländebedingungen noch keineswegs ausgeschöpft. Die in dieser Arbeit präsentierten stellen dahingehend erst den Anfang dar.

7.7 Autonomes Laufen in natürlichem Gelände

Damit eine Laufmaschine ein Aufgabenszenario wie in Abschnitt 1.2 beschrieben autonom bewältigen kann, müssen neben den vorgeschlagenen Navigationsalgorithmen noch zwei wichtige Bindeglieder realisiert werden, die im Folgenden kurz vorgestellt werden. Anschließend werden die Fähigkeiten des Gesamtsystems bei der Durchführung eines Auftrags in natürlichem Gelände unter Beweis gestellt.

7.7.1 Bahnverfolgung

Im bisherigen Verlauf der Arbeit wurde vorgestellt wie die Laufmaschine ihre Lage im Raum ermittelt, wie ein Modell der Umwelt aufgebaut wird, auf dessen Basis dann eine Bahn zum Ziel hin geplant wird. Ein wichtiges Bindeglied zur reaktiven Steuerung ist die Bahnverfolgung. Diese hat dafür Sorge zu tragen, dass der von der Bahnplanung

vorgegebene Weg vom Roboter auch eingehalten wird. Die Bahnverfolgung gibt somit die aktuell vom Roboter einzustellende Bewegungsrichtung vor: Soll der Roboter geradeaus laufen, sich drehen oder doch eher seitwärts laufen? Die Geschwindigkeit ist hierbei nicht primär von Interesse. Diese wird durch die Adaption automatisch an die aktuellen Gegebenheiten angepasst.

7.7.2 Auftragsverwaltung

Die Auftragsverwaltung nimmt den Auftrag mit den Zielkoordinaten vom Leitstand entgegen und sorgt für einen geordneten Ablauf der Navigationsalgorithmen. Die realisierte Steuerungsinstanz unternimmt die folgenden Schritte:

Initiale Datenaufnahme: Geht der erste Auftrag ein, liegen in der Regel noch keinerlei Umweltinformationen vor. Deshalb wird zunächst eine Datenaufnahme initiiert. Der Laserscanner wird mit dem Sensorkopf einmal auf- und abbewegt, um die unmittelbare Umgebung zu kartieren. Anschließend wird eine erste Bahn von der aktuellen Position hin zum Zielort geplant. Diese wird von der Bahnverfolgung in Form von Richtungsvorgaben an die Adaption stückweise weitergeleitet. Schließlich leitet die Adaption die an die Umgebung angepassten Steuerungsparameter an die reaktive Laufmaschinensteuerung weiter.

Überprüfung des Plans: Auf Grund der begrenzten Sensorreichweite ist zum Planungszeitpunkt die Karte nicht vollständig bekannt. Des Weiteren fließen während des Laufens ständig aktuelle Sensorinformationen ins Umweltmodell ein, so dass der Plan unterwegs überprüft und bei Bedarf verändert werden muss. Diese Überprüfung wird ständig nach einer aktualisierten Gebietsbewertung durchgeführt (vergleiche Abschnitt 7.6).

Auftragsende: Ist der vorgegebene Zielort erreicht, wird die Laufmaschinen angehalten und der Leitstand informiert.

7.7.3 Waldszenario

Das von LAURON IV im Folgenden zu bewältigende Szenario ist in Abbildung 7.28 abgebildet. Die während der initialen Datenaufnahme erzeugte 3D-Karte ist in Abbildung 7.29 dargestellt. Die vermessenen Geometrien der Umgebung werden im 3D-Umweltmodell zufriedenstellend wiedergegeben. Abbildung 7.30 zeigt und beschreibt die hieraus generierte 2^{1/2}D-Karte sowie den vom Bahnplaner ermittelten Weg.

Im weichen Waldboden sinken die Füße von LAURON IV stark in den Untergrund ein, so dass der exakte Zeitpunkt des Bodenkontaktes nicht bestimmbar ist. Dies hat zur Folge, dass die während der Lokalisationsexperimente beobachteten Schwierigkeiten bei der Höhenbestimmung durch die Odometrie enorm verstärkt wurden. In Abbildung 7.31(b) ist der Höhenverlauf während des Testlaufs im Wald aufgetragen. Die Roboterhöhe steigt auf einen Meter an, obwohl kaum ein Höhenunterschied zwischen Start- und Zielpunkt besteht.

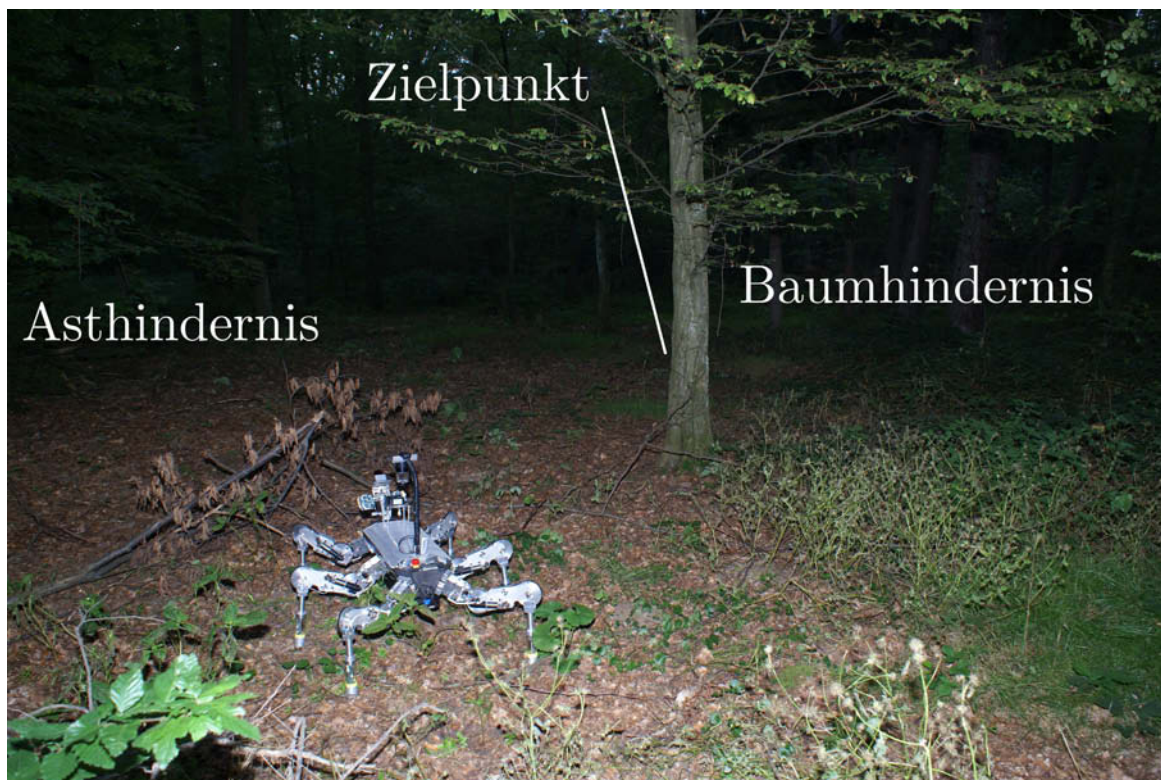


Abbildung 7.28: Waldszenario: Foto von LAURON IV in einem Waldgebiet. Direkt vor dem Roboter liegt ein nicht überwindbarer Ast, auf der rechten Seite steht ein Baum. LAURON IV hat die Aufgabe, den vorgegebenen Zielpunkt autonom zu erreichen.

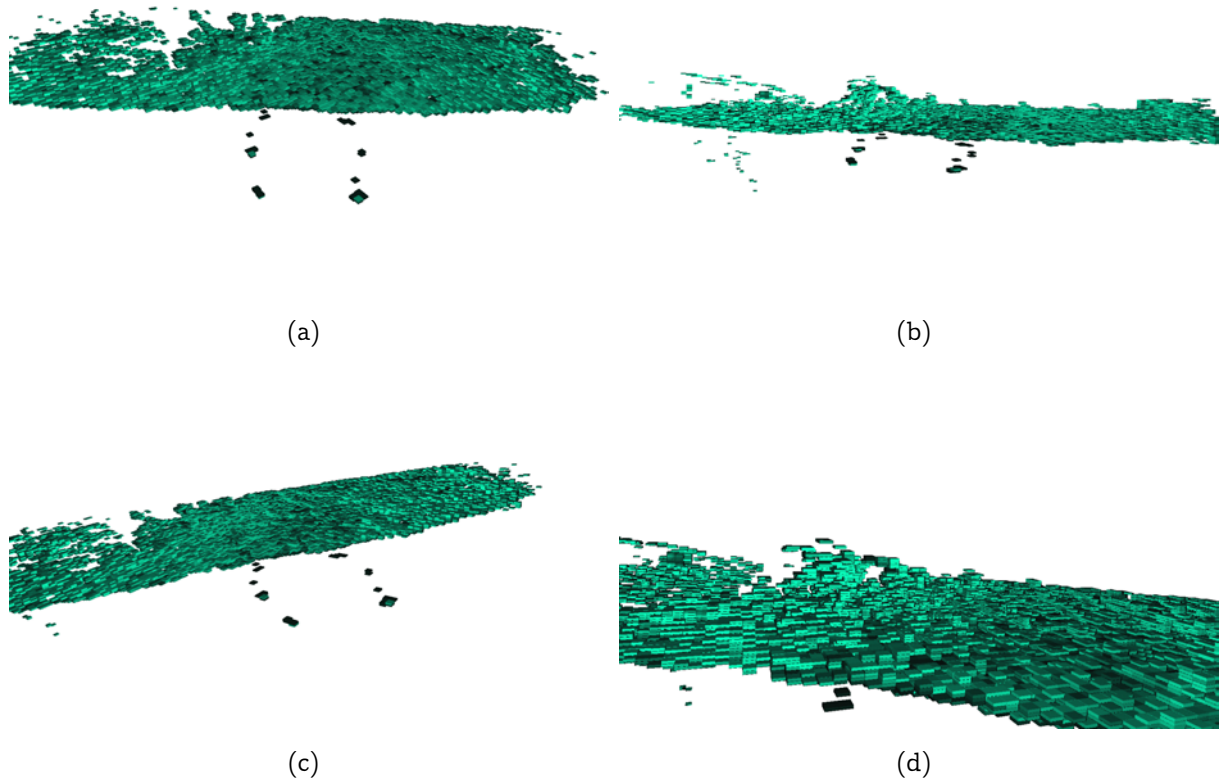


Abbildung 7.29: Waldszenario: Verschiedene Ansichten der dreidimensionalen Urkarte nach der initialen Datenaufnahme. (a) Sicht von schräg oben. In der unteren Bildhälfte sind die sechs einzelnen Fußpunkte der Laufmaschine zu erkennen. (b) In der flachen Ansicht treten die Erhebungen des Astes direkt vor LAURON IV (in der Mitte des Bildes) und des Baumes (am rechten Bildrand) hervor. (c) Sicht von schräg oben in Richtung des Baumes neben dem Roboter. Da der Laser beim initialen Scan nicht über die Waagrechte hinaus bewegt wurde, ist die reale Höhe des Baumes in der Karte nicht erfasst. (d) In der Nahaufnahme aus der Perspektive des Roboters auf den Ast ist eine deutliche Lücke zu erkennen. Der Laser hat durch eine Astgabel hindurch kein Material detektieren können.

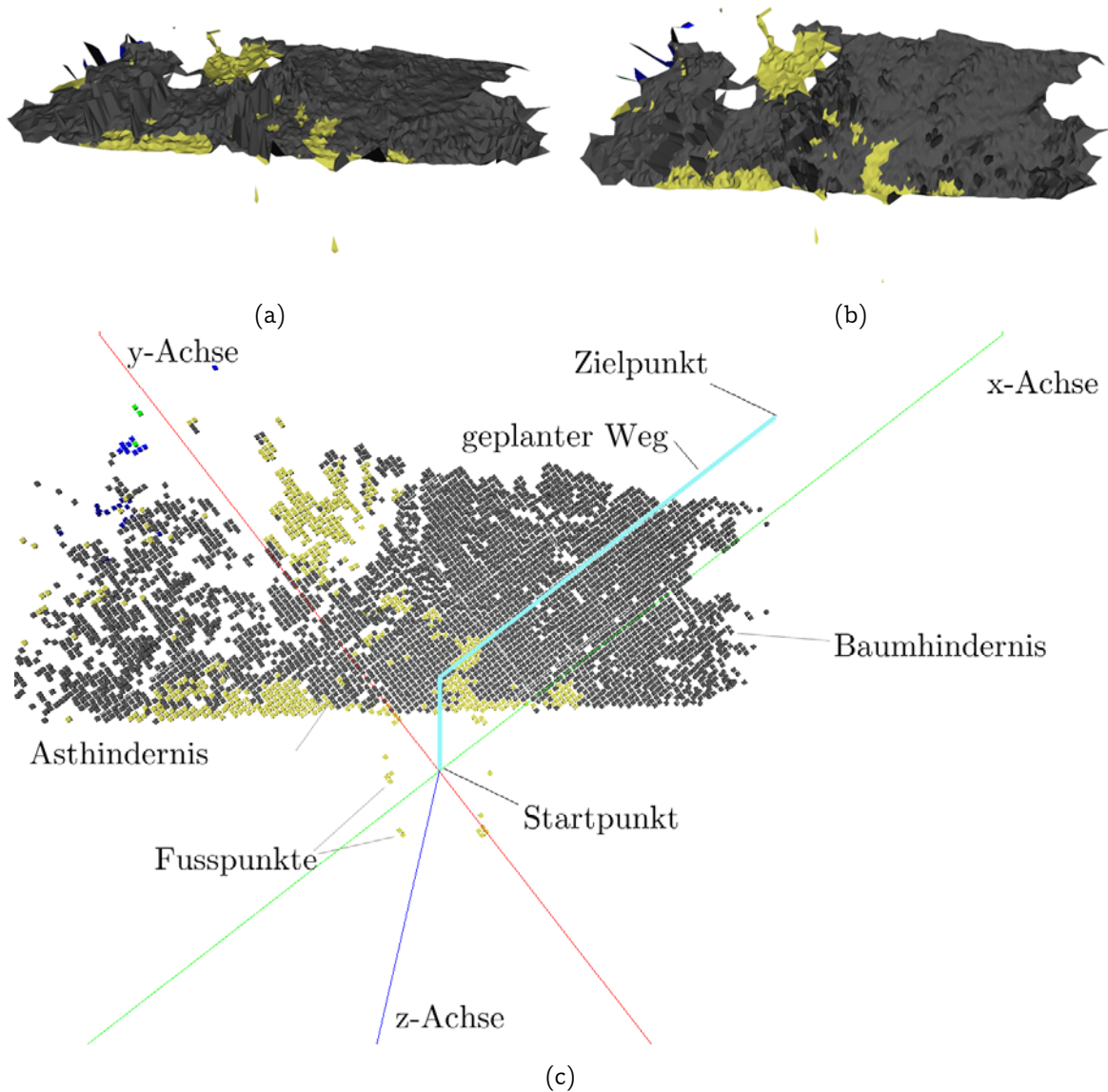


Abbildung 7.30: Waldszenario: Verschiedene Ansichten der $2\frac{1}{2}$ D-Karte nach der initialen Datenaufnahme. (a)(b) Ansichten der $2\frac{1}{2}$ D-Karte mit Hilfe der Delaunay-Triangulierung. Da einige weiter entfernten Punkte (oben rechts im Bild) tiefer gelegen sind, wird der Großteil der Karte durch die Visualisierungssoftware in einer Farbe dargestellt. (c) In der Draufsicht sind neben dem globalen Koordinatensystem die wesentlichen bisher beschriebenen Orte sowie der geplante Weg eingezeichnet. Dem Startpunkt der Planung entspricht der Körpermitte des Roboters und liegt im Koordinatenursprung. Die Ausdehnung der Laufmaschine zeichnet sich durch die sechs umliegenden Beine ab. Der Ast wird von der positiven y-Achse geschnitten. Der erwähnte Baum liegt am rechten Kartenrand knapp unter der x-Achse. Der Zielpunkt liegt vom Roboter aus gesehen schräg hinter dem Baum. Auf direktem Weg vom Start- zum Zielpunkt würde die Laufmaschine mit dem Baum kollidieren. Die Bahnplanung legt den Pfad in sicherer Entfernung am Baumhindernis vorbei.

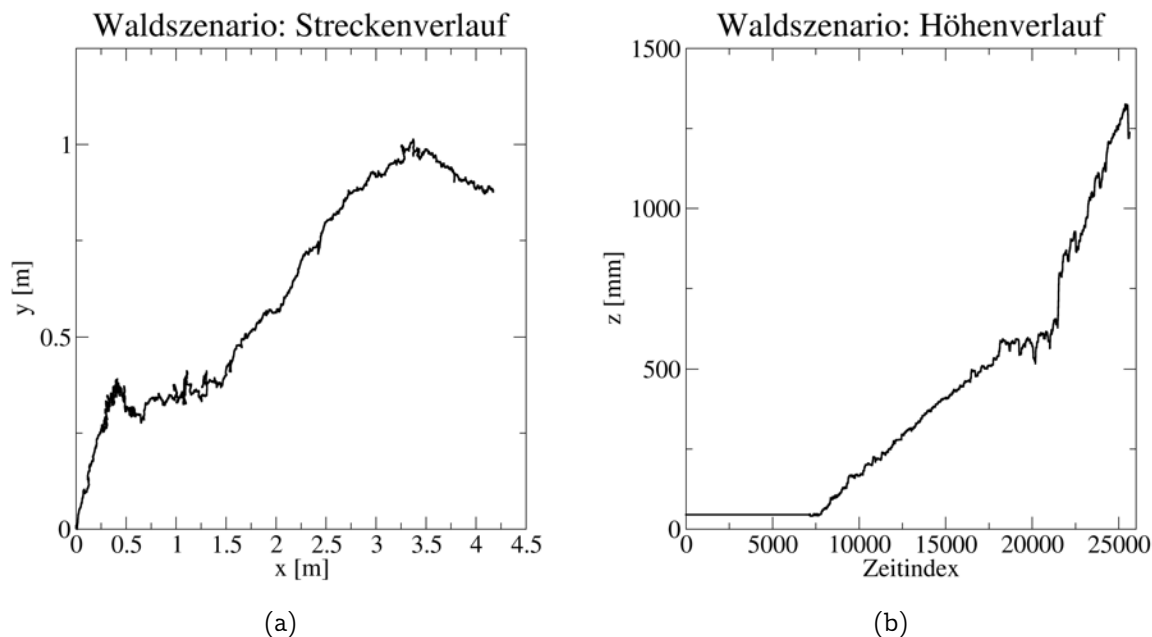


Abbildung 7.31: Waldszenario: Der während des Versuchslaufs durch die Lokalisation ermittelte (a) Streckenverlauf und (b) Höhenverlauf. Die Roboterhöhe steigt ständig an, obwohl sich die Laufmaschinen im Wald in etwa auf einer Ebene bewegt.

Da der Sensorkopf während des Laufens immer nur den Bereich direkt vor dem Roboter abscannt, um ständig aktuelle Daten für die Fußaufsetzpunktplanung zur Verfügung zu stellen, bildet sich durch den Umstand der steigenden Roboterhöhe im Umweltmodell ein Abhang aus (vergleiche Abbildung 7.32). Dies hat zur Folge, dass der bisher geplante Weg zum Zielort für den Roboter nicht mehr begehbar ist.

Darüberhinaus bewirkt die im Umweltmodell verzeichnete große Schräge, dass die Adaption den Roboter sehr stark abbremst, dessen Zentralkörper absenkt, einen breiten Stand wählt und sich parallel zum Untergrund ausrichtet. In dieser Konfiguration kam LAURON IV auf dem Waldboden kaum mehr voran. An dieser Stelle musste sowohl die automatische Adaption an das Gelände als auch die Bahnverfolgung abgeschaltet werden. Den restlichen Weg zum Zielort wurde der Roboter entlang des ursprünglich geplanten Pfades manuell gesteuert.

7.8 Zusammenfassung und Ergebnisse

Die durchgeführten Experimente belegen, dass sich die in dieser Arbeit entwickelten Algorithmen und Methoden zur Navigation von Laufmaschinen eignen. Um der Laufmaschine dies zu ermöglichen, muss deren Sensorsystem entsprechend erweitert werden.

Sind der Auftrittszeitpunkt und die Lage der Beine im Roboterkoordinatensystem hinreichend genau bekannt, können hieraus odometrische Lageinformationen mit einer Genauigkeit von rund 5–10% berechnet werden. Da die Odometrie ein relativ messendes Verfahren darstellt, können die auftretenden Fehler unbegrenzt wachsen, so dass zu deren Beschränkung Korrekturen durchgeführt werden müssen. Das in dieser Ar-

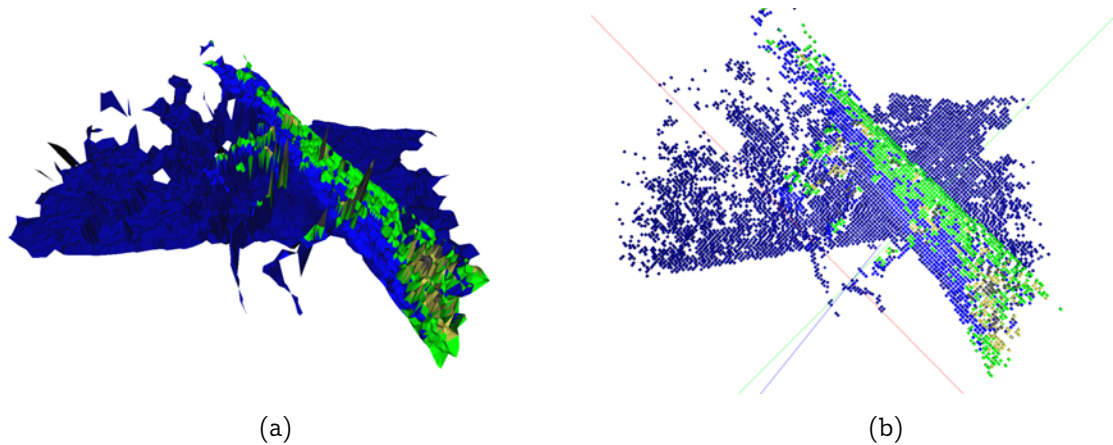


Abbildung 7.32: Waldszenario: Ansicht der veränderten Karten nachdem der Roboter eine Teilstrecke überwunden hat: (a) Delaunay-Triangulierung (b) Draufsicht. Durch eine Drift der Roboterlage in z-Richtung, werden aktuelle Messungen im Vergleich zur initialen Messung an zu hoher Stelle in der Karte eingetragen. Der ursprüngliche Weg kann nicht mehr beschritten werden.

beit entwickelte Kalman-Filter kann zur optimalen Fusion absoluter Lageinformationen herangezogen werden.

Die Bestimmung der absoluten Roboterneigung bezüglich der Erde kann mit Hilfe von Beschleunigungssensoren sicher und korrekt erfolgen. Der Einsatz von Magnetfeldsensoren zur Ermittlung der Nordrichtung hat sich jedoch als zu störanfällig herausgestellt, um korrekte globale Karten aufzubauen. Für kürzere Missionsdauern können hier beispielsweise genaue faseroptische Gyroskope Abhilfe schaffen. Sie erfassen die Orientierung nur relativ, jedoch mit einer geringen Drift, dass auch globale Karten kaum darunter leiden würden. Begnügt man sich mit lokal korrekten Karten, kann man mit kalibrierten Magnetfeldsensoren auskommen. Bei entsprechend aufwändiger Kalibrierung könnten – eine entsprechende Kalibrierungsumgebung vorausgesetzt – die in dieser Arbeit erzielten Ergebnisse noch verbessert werden.

Kostengünstige GPS-Empfänger haben sich zur Bestimmung der absoluten Position der Laufmaschine im Raum als ungeeignet herausgestellt. Die empfangenen GPS-Daten sind viel zu ungenau. Sie würden den Aufbau des Umweltmodells stören. Sind genaue absolute Positionsangaben für die Missionsdurchführung unabdingbar, muss auf zentimetergenaue GPS-Empfänger zurückgegriffen werden.

Die Kartierungsexperimente haben gezeigt, dass zumindest lokal korrekte Karten mit der Odometrie und einer „verzerrten“ Orientierungskorrektur möglich sind. Lokal korrekt soll in diesem Zusammenhang heißen: die direkt vor dem Roboter liegende Umwelt wird durch den Einsatz der Umweltsensorik zumeist korrekt im eigens hierfür entworfenen erweiterten Inferenzgitter eingetragen. Die daraus erzeugte Höhenkarte stimmt in der unmittelbaren Roboterumgebung zum Großteil mit der Realität über ein. Und sollten die Planungsebenen auf Grund verzerrter Karten doch das ein oder andere Mal falsche Entscheidungen treffen, so sollte die darunterliegende reaktive Steuerung noch immer mit den Ungenauigkeiten umgehen können.

Die Eignung des entworfenen Bahnplaners wurde durch mehrere Einzelexperimente

unter Beweis gestellt. Dessen bessere Eignung für Laufmaschinen als andere in der Literatur beschriebene Ansätze wurde dabei besonders unterstrichen.

Schließlich wurde gezeigt, dass das Laufverhalten des Roboters durch Einsatz des gesammelten Umweltwissens nachhaltig verbessert werden kann, indem die Laufparameter automatisch an die aktuell vorherrschenden Gegebenheiten angepasst werden.

Durch die Zusammenführung der einzelnen Teilbereiche zu einem kompletten Navigationssystem für Laufroboter konnten die beschriebenen Fähigkeiten auch in natürlichem Gelände bestätigt werden. Die vorgeschlagene Methode zur Umweltmodellierung bildet natürliches Gelände ausreichend genau ab, um damit sowohl eine Bahnplanung als auch eine automatische Anpassung an den Untergrund zu ermöglichen. Es war jedoch auch zu sehen, dass die Lokalisationsschwierigkeiten sich in natürlichem Gelände derart verstärken können, dass die Gesamtaufgabe nicht mehr lösbar ist.

Kapitel 8

Zusammenfassung der wesentlichen Aspekte des Forschungsziels und Ausblick auf fortführende Arbeiten

Die Zielsetzung der vorliegenden Arbeit war die Erforschung von Navigationsstrategien, die es einer Laufmaschine im Gelände ermöglichen, ein vorgegebenes Ziel autonom zu erreichen. Die in dieser Arbeit zur Erreichung dieses Forschungsziels geleisteten wissenschaftlichen Beiträge werden im Folgenden zusammengefasst.

Lokalisation

Um der Laufmaschine eine stabile und ständig verfügbare Methode zur Lagebestimmung zur Verfügung zu stellen, wurde ein in der Literatur beschriebenes Verfahren zur Berechnung der Odometrie von Laufmaschinen für den allgemeinen Einsatz auf Laufmaschinen erweitert. Hierzu wurde ein auf Methoden der Fuzzy-Logik basierendes Bewertungsverfahren zur Bestimmung von Beingewichten entwickelt. Die Genauigkeit der Odometrie wurde bei Experimenten auf dem Laufroboter LAURON IV evaluiert.

Da schon kleine Fehler relativer Lokalisationsverfahren zu unbegrenzt wachsenden Lokalisationsfehlern führen können, wurde ein Kalman-Filter-Modell entwickelt, um Positionsinformationen absolut messender Sensoren optimal mit der Lage aus der Odometrieberechnung zu fusionieren. Das an die speziellen Anforderungen von Laufmaschinen angepasste Systemmodell macht keinerlei Annahmen über die Roboterdynamik. Darüberhinaus passt es sich dynamisch an die aktuellen Gegebenheiten an. Dies eröffnet dessen allgemeine Anwendbarkeit über den Bereich von Laufmaschinen hinaus. Obgleich die Güte der auf LAURON IV eingesetzten absoluten Sensoren teilweise nicht den an sie gestellten Anforderungen genügten, konnte die Korrektheit des konzipierten Filter-Modells nachgewiesen werden.

Umweltmodellierung

Die Randbedingungen für den Einsatz auf autonomen Laufmaschinen im Gelände, machten es erforderlich, ein neues Verfahren zur Erstellung dreidimensionaler geome-

trischer Karten zu entwickeln. Das erweiterte Inferenzgitter ist eine Variante eines Belegtheitsgitters. Neben einer Belegtheitswahrscheinlichkeit wird zusätzlich ein Zuverlässigkeitswert in der einzelnen Gitterzelle gespeichert, um sowohl den Zustand des Geländes als auch die Sicherheit über diese Information zu modellieren. Um effizient neue Messungen in das Gitter einzutragen, wird auf eine komplexe mathematische Modellierung verzichtet und stattdessen eine auf Methoden der Fuzzy-Logik basierende Fusionsmethode eingesetzt. Diese ermöglicht es, auf einfache Weise die unterschiedlichsten Einflussgrößen wie Sensorqualität, Anzahl bisheriger Messungen oder eine Nachbarschaftsbeziehung mit filternden Eigenschaften zu modellieren und zu berechnen. Die Leistungsfähigkeit dieses Verfahrens wurde durch Experimente bekräftigt.

Um trotz des dreidimensionalen Modells eine Planung in Echtzeit zu ermöglichen, wird parallel zum Aufbau des 3D-Modells eine zweidimensionale Höhenkarte ($2\frac{1}{2}$ D-Karte) generiert. Diese kann ständig aktuell gehalten werden, ohne den Aufwand bei jeder Messung ins Unermessliche zu steigern. Auch die Korrektheit dieses Verfahrens wurde experimentell bestätigt.

Bahnplanung

Im Bereich der Bahnplanung für Laufmaschinen lassen sich in der Literatur zwar durchaus einige Verfahren finden, doch werden diese nicht allen in dieser Arbeit gestellten Anforderungen gerecht. Deshalb musste unter Berücksichtigung der Vor- und Nachteile bestehender Verfahren auch hier ein eigener Ansatz verfolgt werden. Ausgehend von einem generischen Laufmaschinenmodell, das beispielsweise auch die Körperneigung und zumindest rudimentär die Beingeometrie berücksichtigt, wird jede für die Planung erforderliche Gitterzelle des Umweltmodells einer stufenweise aufgebauten Bewertungsfunktion zugeführt. Deren Ergebnis wird zum Aufbau eines abstoßenden Potentials genutzt, das in Kombination mit einem vom Ziel ausgehenden anziehenden Potential ein Potentialfeld bildet, das anschließend von der Wegsuche verwendet wird. Weiterhin kann das Verfahren der vorliegenden Arbeit mit unbekanntem Gebieten umgehen, was insbesondere für den praktischen Einsatz unverzichtbar ist. In unterschiedlichen Experimenten hat sich gezeigt, dass dieser Bahnplaner allen Anforderungen und Randbedingungen gewachsen ist.

Adaption von Laufverhalten

Um zu zeigen, welche weitergehenden zukünftigen Möglichkeiten der Einsatz eines geometrischen Umweltmodells bietet, wurden schließlich noch zwei Verfahren entwickelt, die das Laufverhalten des Roboters an die aktuellen Umgebungsbedingungen anpassen. Auf Basis der geometrischen Höhenkarte wird eine Bewertung des von der Laufmaschine gerade durchlaufenen Gebietes vorgenommen. Mit Hilfe unscharfer Regeln und dem nötigen Expertenwissen konnte darauf aufbauend eine Regelbasis definiert werden, die eine automatische Adaption von Laufparametern ermöglicht. Des Weiteren wird das im Umweltmodell gespeicherte Wissen um sichere Fußaufsetzpunkte genutzt, um die Schwingbeine der Laufmaschine an einem geeigneten Ort zu platzieren.

Weitere Arbeiten

Aufgrund der umfangreichen Navigationsthematik konnten in dieser Arbeit nicht alle denkbaren und möglichen Ansätze verfolgt werden. Insbesondere der Bereich der Lokalisation verlangt nach weitergehenden Lösungen. Es mag sein, dass teure Sensorik die in dieser Arbeit aufgetretenen Genauigkeitsprobleme mindern. Vollständig lösen werden auch diese sie nicht. Beispielsweise bleiben die folgenden Fragestellungen zu beantworten:

- Wie kann man die Höhe der Laufmaschine bis auf wenige Zentimeter genau bestimmen?
- Auf welche Positionsinformationen kann man zurückgreifen, wenn kein Sichtkontakt zu Satelliten besteht?
- Wie kann die absolute Orientierung der Laufmaschine robust gegen externe Störungen bestimmt werden?

Vieles deutet darauf hin, dass nur ein ganzes Portfolio unterschiedlicher Verfahren hierzu in der Lage ist. Das in dieser Arbeit entwickelte Kalman-Filter könnte hierbei zur Fusion der weiteren Sensorsysteme herangezogen werden. Das zur Modellierung der Umwelt konzipierte erweiterte Inferenzgitter bietet ebenfalls die Möglichkeit, weitere Sensorsysteme zur Umwelterfassung zu integrieren.

Eine große Herausforderung stellt der Einsatz von landmarken- und/oder kartenbasierter Lokalisationsverfahren auf Laufmaschinen dar. Sind derartige Ansätze im Innenbereich schon intensiv erforscht, steckt deren Einsatz im Gelände noch in den Anfängen. Mit den zusätzlichen Anforderungen und Randbedingungen von Laufmaschinen erschließt sich ein offenes Feld für eigene Ideen.

In dieser Arbeit gar nicht betrachtet wurden kamerabasierte Verfahren. Einer der direkt nächsten Schritte auf LAURON könnte die Ausnutzung des schon vorhandenen Stereokamerakopfes sein. Kameras können auf allen Ebenen der Navigation von Nutzen sein: Die landmarkenbasierte Lokalisation, das Eintragen von Stereodistanzbildern ins Umweltmodell, das Finden des Zieles oder die visuelle Bewertung des Untergrundes sind nur einige Beispiele hierzu.

Wie schon des öfteren betont, wurden in dieser Arbeit die Adaptionmöglichkeiten an die Umwelt bei weitem nicht ausgeschöpft. Die Ermittlung optimaler Strategien zum Durchqueren von Gelände etwa, das Lösen komplexer Kletteraufgaben oder gar das maschinell unterstützte Erlernen derartiger Strategien zeigen einen immensen Forschungsbedarf auf.

Anhang A

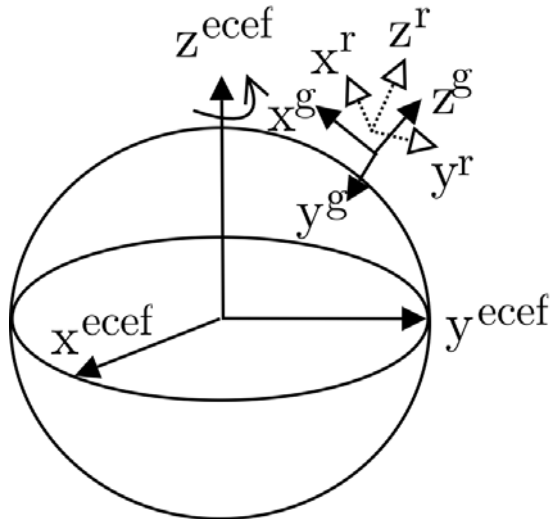
Grundlagen zur Lokalisation

A.1 Koordinatensysteme zur Lokalisation

Soll ein Roboter lokalisiert werden, wird grundsätzlich ein Koordinatensystem benötigt, bezüglich dessen die Lage des Roboters festzustellen ist. Im Folgenden werden drei Koordinatensysteme vorgestellt, die für die Lokalisation in dieser Arbeit Verwendung finden: Ein globales, erdfestes Koordinatensystem, in dem die Position auf der Erde angegeben wird, ein lokales, erdfestes Koordinatensystem, in dem die Position relativ zu einem bestimmten, festen Punkt angegeben wird, sowie ein roboterfestes Koordinatensystem, in dem die Sensorwerte liegen. Die Integration der Messungen findet in einem lokalen, erdfesten Horizontsystem statt.

Erdfestes, globales Weltkoordinatensystem: Soll eine globale Position auf der Erde bestimmt werden, so wird häufig ein erdfestes, globales Koordinatensystem genutzt, auch als *earth-centered earth-fixed* (ecef) Koordinatensystem bezeichnet. Das WGS84-Koordinatensystem ist ein solches erdfestes Koordinatensystem und wird durch das GPS-System unterstützt. Es charakterisiert jede Position eindeutig durch Längengrad, Breitengrad und Höhe über der Erde. Der Ursprung des ecef-Koordinatensystems ist im Erdmittelpunkt verankert. Die z-Achse z^{ecef} zeigt in Richtung der Rotationsachse der Erde. Die x-Achse x^{ecef} und die y-Achse y^{ecef} stehen senkrecht aufeinander und liegen in der Äquatorebene (Abbildung A.1).

Erdfestes, lokales Horizontalsystem: Interessieren kleinere metrische Distanzen, so arbeitet man oft in einem erdfesten Horizontalsystem. Hierzu wird die durch x- und y-Achse aufgespannte Ebene als Tangentialebene an das geodätische Referenzellipsoid gelegt, welches zur Beschreibung der Erde benutzt wird. Dieses Koordinatensystem wird auch als *local tangent plane* (ltp) Koordinatensystem bezeichnet. In dieser Arbeit stellt es das *globale Koordinatensystem* (g) dar, in dem der Roboter agiert. Der Ursprung dieses ebenfalls erdfesten Koordinatensystems wird fest durch Längen- und Breitengrad definiert. Das ltp-Koordinatensystem bzw. g ist hier so definiert, dass die x-Achse x^g nordwärts, die y-Achse y^g westwärts und die z-Achse z^g nach oben zeigt (Abbildung A.1). Die Transformation vom ecef-Koordinatensystem, in dem die GPS-Daten liegen, in ein ltp-Koordinatensystem



ecef: erdfestes, globales
Weltkoordinatensystem

g: erdfestes, lokales
Horizontalsystem

r: Roboterkoordinatensystem

Abbildung A.1: Illustriert die Lage der drei beschriebenen Koordinatensysteme zueinander.

ist in Anhang A.3.2 aufgeführt. Im Gegensatz zum *north-east-down* Navigationskoordinatensystem, das häufig in der Luft- und Raumfahrt verwandt wird, bewegt sich das hier definierte ltp-Koordinatensystem nicht mit dem Roboter mit. Somit spielen bei der späteren Modellierung Coriolis-Kräfte oder Abhängigkeiten vom Breitengrad keine Rolle.

Roboterkoordinatensystem: Der Ursprung und die Achsen des *Roboterkoordinatensystems* (r) sind fest mit dem Roboter verbunden und bewegen sich mit diesem mit. Hierbei zeigt die x -Achse x^r in die Vorwärtsbewegungsrichtung des Roboters und die y -Achse y^r zeigt nach links. Die z -Achse z^r steht orthogonal auf den beiden anderen Achsen und ist aufwärts gerichtet. Diese Achsendefinitionen sind jedoch nicht eindeutig. Beispielsweise werden in Luftfahrtanwendungen oft andere Achsenrichtungen verwendet als in Roboteranwendungen. In Abbildung A.1 ist ein Roboterkoordinatensystem relativ zu einem ltp-Koordinatensystem dargestellt.

Alle Informationen über die beschriebenen Koordinatensysteme wurden (Grewal et al., 2001) und (Farrell und Barth, 1998) entnommen.

A.2 Repräsentation der Lage

Zur Definition einer Lage L im dreidimensionalen Raum wird sowohl die dreidimensionale Position $\mathbf{p} = (x, y, z)^T$ als auch die Orientierung $\boldsymbol{\psi}$ des Körpers bezüglich des Referenzkoordinatensystems benötigt. Jede beliebige Orientierung im Raum lässt sich durch drei unabhängige Rotationen $\boldsymbol{\psi} = (\alpha, \beta, \gamma)^T$ um die Koordinatenachsen darstellen. Diese Rotationen werden oft – in Anlehnung an entsprechende Flugmanöver – als *roll* (rollen), *pitch* (neigen) und *yaw* (gieren) bezeichnet. Somit kann jede Lage im Raum durch ein Sechs-Tupel $(x, y, z, \alpha, \beta, \gamma)$ eindeutig repräsentiert werden. Hierbei ist

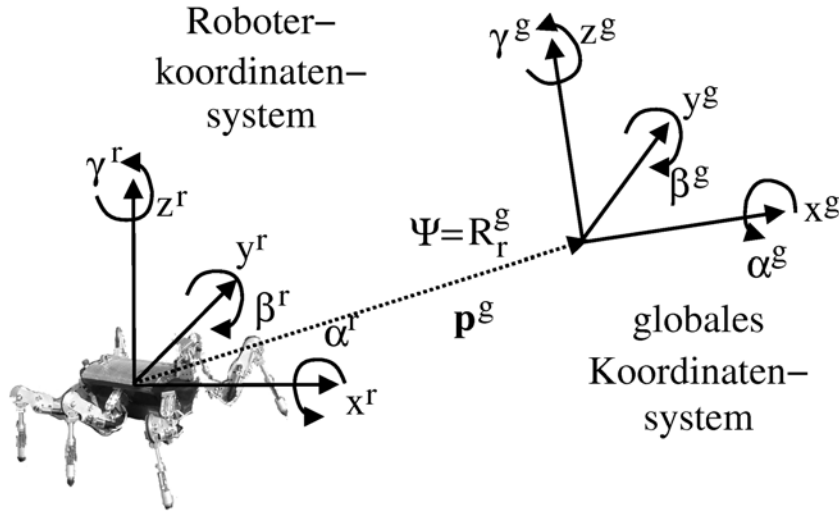


Abbildung A.2: Illustriert die Lage $(\mathbf{p}, \boldsymbol{\psi})^T$ des Roboterkoordinatensystems (r) von LAURON in Bezug auf das globale Koordinatensystem (g).

zu beachten, dass die Bedeutung der drei Rotationswinkel α , β und γ eindeutig definiert sein muss. Je nachdem in welcher Reihenfolge und um welche Achsen gedreht wird, ergibt sich eine unterschiedliche Orientierung im Raum (vergleiche Abbildung A.2).

In dieser Arbeit gilt das Folgende: Ein Körper wird zuerst um den Winkel α um die x -Achse, dann um den Winkel β um die y -Achse und schließlich um den Winkel γ um die z -Achse jeweils des Referenzkoordinatensystems gedreht.

Jede der drei Rotationen lässt sich als 3×3 Rotationsmatrix darstellen:

$$R_\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \quad (\text{A.1})$$

$$R_\beta = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \quad (\text{A.2})$$

$$R_\gamma = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{A.3})$$

Da die Rotationen hintereinander ausgeführt werden, erhält man durch deren Verkettung die folgende Gesamtrrotationsmatrix R :

$$R = R_\alpha R_\beta R_\gamma = \begin{pmatrix} \cos(\beta) \cos(\gamma) & & & \\ \cos(\alpha) \sin(\gamma) + \sin(\alpha) \sin(\beta) \cos(\gamma) & & & \\ \sin(\alpha) \sin(\gamma) - \cos(\alpha) \sin(\beta) \cos(\gamma) & & & \\ & -\cos(\beta) \sin(\gamma) & \sin(\beta) & \\ \cos(\alpha) \cos(\gamma) - \sin(\alpha) \sin(\beta) \sin(\gamma) & -\sin(\alpha) \cos(\beta) & & \\ \sin(\alpha) \cos(\gamma) + \cos(\alpha) \sin(\beta) \sin(\gamma) & \cos(\alpha) \cos(\beta) & & \end{pmatrix} \quad (\text{A.4})$$

Umgekehrt kann man aus einer gegebenen Rotationsmatrix R die Rotationswinkel α , β und γ berechnen. Mit gegebener Gesamtrrotationsmatrix

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (\text{A.5})$$

lässt sich (A.4) schrittweise umformen:

$$R = R_\alpha R_\beta R_\gamma \quad (\text{A.6})$$

$$RR_\gamma^{-1} = R_\alpha R_\beta \quad (\text{A.7})$$

Da für Rotationsmatrizen $R^{-1} = R^\top$ gilt, wird dies zu:

$$RR_\gamma^\top = R_\alpha R_\beta \quad (\text{A.8})$$

Ausmultipliziert ergibt sich die folgende Gleichung:

$$\begin{pmatrix} r_{11} \cos(\gamma) - r_{12} \sin(\gamma) & r_{11} \sin(\gamma) + r_{12} \cos(\gamma) & r_{13} \\ r_{21} \cos(\gamma) - r_{22} \sin(\gamma) & r_{21} \sin(\gamma) + r_{22} \cos(\gamma) & r_{23} \\ r_{31} \cos(\gamma) - r_{32} \sin(\gamma) & r_{31} \sin(\gamma) + r_{32} \cos(\gamma) & r_{33} \end{pmatrix} = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ \sin(\alpha) \sin(\beta) & \cos(\alpha) & -\sin(\alpha) \cos(\beta) \\ -\cos(\alpha) \sin(\beta) & \sin(\alpha) & \cos(\alpha) \cos(\beta) \end{pmatrix} \quad (\text{A.9})$$

Das entspricht den folgenden neun Einzelgleichungen:

$$\cos(\beta) = r_{11} \cos(\gamma) - r_{12} \sin(\gamma) \quad (\text{A.10})$$

$$0 = r_{11} \sin(\gamma) + r_{12} \cos(\gamma) \quad (\text{A.11})$$

$$\sin(\beta) = r_{13} \quad (\text{A.12})$$

$$\sin(\alpha) \sin(\beta) = r_{21} \cos(\gamma) - r_{22} \sin(\gamma) \quad (\text{A.13})$$

$$\cos(\alpha) = r_{21} \sin(\gamma) + r_{22} \cos(\gamma) \quad (\text{A.14})$$

$$-\sin(\alpha) \cos(\beta) = r_{23} \quad (\text{A.15})$$

$$-\cos(\alpha) \sin(\beta) = r_{31} \cos(\gamma) - r_{32} \sin(\gamma) \quad (\text{A.16})$$

$$\sin(\alpha) = r_{31} \sin(\gamma) + r_{32} \cos(\gamma) \quad (\text{A.17})$$

$$\cos(\alpha) \cos(\beta) = r_{33} \quad (\text{A.18})$$

Aus (A.11) erhält man:

$$\gamma = \text{atan2}(-r_{12}, r_{11}) \quad (\text{A.19})$$

mit (A.10) und (A.12) ergibt sich:

$$(\text{A.20})$$

$$\beta = \text{atan2}(r_{13}, r_{11} \cos(\gamma) - r_{12} \sin(\gamma)) \quad (\text{A.21})$$

und schließlich berechnet sich aus (A.14) und (A.17) noch:

$$(\text{A.22})$$

$$\alpha = \text{atan2}(r_{31} \sin(\gamma) + r_{32} \cos(\gamma), r_{21} \sin(\gamma) + r_{22} \cos(\gamma)) \quad (\text{A.23})$$

A.3 Koordinatentransformationen

In diesem Abschnitt werden verschiedene Koordinatentransformationen aufgeführt, die in dieser Arbeit gebraucht wurden. Sie wurden u.a. (Farrell und Barth, 1998) entnommen.

A.3.1 Innerhalb des ecef -System

Im ecef-Koordinatensystem kann ein Punkt in geodätischen ecef-Koordinaten (Länge, Breite, Höhe), oder in rechtwinkligen ecef-Koordinaten (x, y, z) angegeben werden. Die geodätischen Koordinaten sind an ein Polarkoordinatensystem angelehnt. Im Folgenden wird alles für die Transformation von geodätischen Koordinaten zu rechtwinkligen Koordinaten nötige aufgeführt. Als Erdmodell wird der WGS-84 Ellipsoid benutzt, der wie folgt definiert ist.

$$\text{Gro\ss e Halb achse: } a = 6378137,0 \text{ m} \quad (\text{A.24})$$

$$\text{Kleine Halb achse: } b = 6356752,3142 \text{ m}$$

$$f = \frac{a - b}{a} = 0,0034 \quad (\text{A.25})$$

$$e = \sqrt{f(2 - f)} = 0,0818$$

$$N(\lambda) = \frac{a}{\sqrt{1 - e^2 \sin^2(\lambda)}}$$

Unter Benutzung obiger Variablen lassen sich x, y, z im ecef-System wie folgt berechnen. λ bezeichnet die geodätische Breite, α bezeichnet die geodätische Länge.

$$x = (N + h) \cos(\lambda) \cos(\alpha) \quad (\text{A.26})$$

$$y = (N + h) \cos(\lambda) \sin(\alpha)$$

$$z = [N(1 - e^2) + h] \sin(\lambda)$$

Die inverse Transformation vom rechtwinkligen Koordinatensystem zum geodätischen wurde hier nicht aufgeführt. Sie ist komplizierter und wird iterativ berechnet. Während dieser Arbeit wurde nur die oben aufgeführte Transformation benötigt.

A.3.2 Von ecef nach ltp

GPS-Empfänger liefern die Positionsdaten entweder als (x, y, z) - oder als (λ, α, h) -Punkt im ecef-Koordinatensystem. Für die Lokalisation und die Verwendung der Position im

Kalman-Filter (Abschnitt 3.8) wird die Position in einem *local tangent plane* ltp-Koordinatensystem benötigt (s.o.). Ein ecef-Punkt $(x, y, z)_{\text{ecef}}^T$ wird wie folgt in ltp-Koordinaten $(x, y, z)_{\text{ltp}}^T$ transformiert.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{\text{ltp}} = R_{\text{ecef}}^{\text{ltp}} \left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{\text{ecef}} - \begin{pmatrix} x_u \\ y_u \\ z_u \end{pmatrix}_{\text{ecef}} \right) \quad (\text{A.27})$$

Durch $(x_u, y_u, z_u)_{\text{ecef}}^T$ sind die ecef-Koordinaten des Ursprungs des ltp-Koordinatensystems gegeben. Die Transformationsmatrix $R_{\text{ecef}}^{\text{ltp}}$ wird mittels des Längengrades λ und des Breitengrades α des Koordinatensystemursprungs definiert als:

$$R_{\text{ecef}}^{\text{ltp}} = \begin{pmatrix} -\sin(\lambda) \cos(\alpha) & -\sin(\lambda) \sin(\alpha) & \cos(\lambda) \\ \sin(\alpha) & -\cos(\alpha) & 0 \\ \cos(\lambda) \cos(\alpha) & \cos(\lambda) \sin(\alpha) & \sin(\alpha) \end{pmatrix} \quad (\text{A.28})$$

Hierbei wurde angenommen, dass die z-Achse des ltp-Koordinatensystem nach oben zeigt. Die x-Achse zeigt in Richtung Norden und die y-Achse in Richtung Westen.

A.4 Mehrdimensionale Taylorentwicklung

Die mehrdimensionale Verallgemeinerung der Taylor-Entwicklung hat die Form

$$\begin{aligned} f(\mathbf{x}) &= \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{i_1, \dots, i_n}^D \left[\frac{d^n f}{dx_{i_1} \dots dx_{i_n}} \Big|_{\mathbf{x}} \prod_{j=1}^n (x_{i_j} - X_{i_j}) \right] \\ &= f(\mathbf{X}) + \sum_i^D \frac{df}{dx_i} \Big|_{\mathbf{x}} (x_i - X_i) \\ &\quad + \frac{1}{2} \sum_{i,j}^D \frac{d^2 f}{dx_i dx_j} \Big|_{\mathbf{x}} (x_i - X_i)(x_j - X_j) + O(\mathbf{x} - \mathbf{X})^3 \end{aligned} \quad (\text{A.29})$$

wobei \mathbf{X} der Entwicklungspunkt und D die Dimension des Vektors \mathbf{x} ist. $O(\mathbf{x} - \mathbf{X})^3$ bezeichnet das Restglied einer Taylorentwicklung, die nach dem 2. Glied abgebrochen wurde. Mit diesem Restglied kann man testen, in welchem Konvergenzradius die Approximation gültig ist.

Zum besseren Verständnis wird obige Entwicklungsvorschrift an einem einfachen Beispiel demonstriert. Es wird $f(\mathbf{x}) = \sin x \cdot \sin y$ bis zum 2. Taylorglied an der Stelle

$(x_0, y_0) = (0, 0)$ entwickelt.

$$\begin{aligned}
 \sin x \cdot \sin y &= \underbrace{f(0, 0)}_0 + \left[\underbrace{(\cos x \sin y)|_0}_0 \cdot x + \underbrace{(\sin x \cos y)|_0}_0 \cdot y \right] \\
 &+ \frac{1}{2} \left[\underbrace{(\sin x \sin y)|_0}_0 (x \cdot y) + \underbrace{(\cos x \cos y)|_0}_1 (x \cdot y) \right. \\
 &\left. + \underbrace{(\cos x \cos y)|_0}_1 (x \cdot y) + \underbrace{(\sin x \sin y)|_0}_0 (x \cdot y) \right] \tag{A.30} \\
 &= \frac{1}{2} (2 \cdot x \cdot y) \\
 &= x \cdot y
 \end{aligned}$$

Das Restglied wird hier nicht aufgeführt. Es kann (Bronstein und Semendjajew, 1991) entnommen werden und zeigt auf, dass obige Approximation nur für kleine Werte von $(|x|+|y|)$ gültig ist.

Anhang B

Kurzeinführung in die Fuzzy-Logik

As the complexity of systems increases, our ability to make precise and significant statements about their behaviour diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics¹.

(Zadeh, 1973)

Die Regelung komplexer Prozesse mit Hilfe von Unscharfer Logik (*Fuzzy-Logik*), die 1965 von Lotfi A. Zadeh vorgestellt wurde (Zadeh, 1965), ist im Laufe der Zeit zu einer festen Größe in der Regelungstechnik geworden. Zunächst noch mit wenig positiver Resonanz bedacht, konnten in der zweiten Hälfte der 70er Jahre mehrere unscharfe Regelsysteme erfolgreich in der Industrie eingesetzt werden. Ende der 80er Jahre bescherten vor allem japanische Entwicklungen der Unscharfen Logik einen regelrechten Boom, der sich bis in die halb- und unwissenschaftliche Presselandschaft ausbreitete. Ungeachtet der genauso euphorischen wie unrealistischen Erwartungen dieser Zeit spielen Fuzzy-Systeme, zumeist in Kombination mit anderen Methodiken, weiterhin eine unbestritten bedeutende Rolle in wissenschaftlichen und industriellen Projekten.

In diesem Abschnitt sollen zunächst in gebotener Kürze die benötigten Grundlagen erläutert werden. Für eine ausführliche praxisrelevante Darstellung von Fuzzy-Methoden sei auf Kahlert (1995) verwiesen. Eine sehr gute und ausführliche theoretische Aufbereitung des Themas „Unschärfe“ findet sich in Biewer (1997).

B.1 Fuzzy-Mengen und linguistische Terme

Die Unscharfe Logik bietet die Möglichkeit, umgangssprachlich formuliertes „Expertenwissen“ in eine mathematische Form zu bringen, um es anschließend weiter verarbeiten zu können. So kann empirisch gewonnenes, nicht in exakter Form vorliegendes Modellwissen als Grundlage komplexer Regelprozesse dienen.

Eine *Fuzzy-Menge* A über einem Wertebereich X ist definiert als eine geordnete

¹Je komplexer ein System wird, desto weniger sind wir in der Lage, präzise und gleichwohl bedeutsame Aussagen über sein Verhalten zu machen, bis ab einem bestimmten Punkt sich Präzision und Bedeutsamkeit wechselseitig ausschließen.

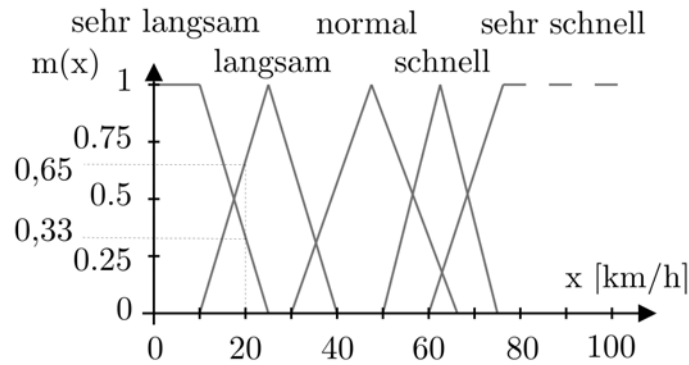


Abbildung B.1: Fuzzy-Mengen am Beispiel der linguistischen Variable $X =$ „Geschwindigkeit im Ortsbereich“: Auf X sind fünf Fuzzy-Mengen definiert, denen die linguistischen Terme *sehr langsam*, *langsam*, *normal*, *schnell* und *sehr schnell* zugeordnet sind. Die Fuzzy-Menge *langsam* beispielsweise überdeckt dabei den Wertebereich von 10 bis 40 km/h. Für den Wert 20 km/h ist eine Fuzzyifizierung skizziert: Sein Zugehörigkeitsgrad zur Fuzzy-Menge *sehr langsam* ist ungefähr 0,33, der zur Menge *langsam* 0,65. Für alle anderen Mengen ist der Zugehörigkeitsgrad 0. Für alle zur linguistischen Variablen gehörenden Fuzzy-Mengen ergibt sich also ein Zugehörigkeitstupel $\mu(20 \text{ km/h}) = (0,33; 0,65; 0; 0; 0)$.

Menge von Paaren

$$A := \{x, \mu_A(x) \mid x \in X\} \quad , \quad (\text{B.1})$$

wobei die Abbildung

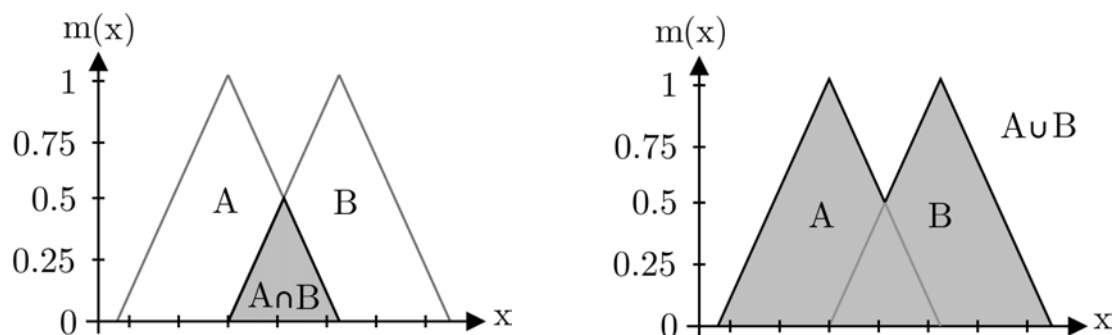
$$\mu_A : X \rightarrow [0, 1], \quad \mu_A \in \mathbb{R} \quad (\text{B.2})$$

jedem Element von X einen *Zugehörigkeitsgrad* $\mu_A(x)$ zuordnet. Diese Funktion heißt *Zugehörigkeitsfunktion* von F . Da eine Fuzzy-Menge durch ihre Zugehörigkeitsfunktion vollständig und eindeutig definiert ist, wird der Einfachheit halber gerne μ_A als Bezeichner für eine Fuzzy-Menge verwendet.

Eine *linguistische Variable* ist eine natürlichsprachliche Größe, beispielsweise „Geschwindigkeit“. Sie wird im Allgemeinen durch eine Reihe so genannter *linguistischer Terme* beschrieben, die Zustände bzgl. der linguistischen Variable umgangssprachlich beschreiben (beispielsweise „langsam“, „schnell“, „sehr schnell“), und denen jeweils eine Fuzzy-Menge zugeordnet ist. Die Gesamtheit der Fuzzy-Mengen überdeckt den gesamten Wertebereich der Variablen².

Bildet man einen *scharfen*, also numerisch wohldefinierten Wert einer linguistischen Variable (in unserem Beispiel könnte das 50 km/h sein) auf die linguistischen Terme ab, indem man für den scharfen Wert den jeweiligen *Zugehörigkeitsgrad* zu jeder Fuzzy-Menge ermittelt, spricht man von *Fuzzyifizierung*. Ein Beispiel dazu wird in Abbildung B.1 erläutert.

²In den Beispielen in diesem Abschnitt ist dies aus Gründen der Übersicht nicht immer so. Man denke sich zur vollständigen Überdeckung des Wertebereichs einfach noch ein paar Fuzzy-Mengen hinzu.



(a) UND-Operator: Schnitt von Fuzzy-Mengen (b) ODER-Operator: Vereinigung von Fuzzy-Mengen

Abbildung B.2: UND- und ODER-Operator auf Fuzzy-Mengen

B.2 Operatoren auf Fuzzy-Mengen

Auf Fuzzy-Mengen kann man, wie auch in klassischen Logiken, logische Operatoren definieren. Es existiert eine Vielzahl unterschiedlicher Möglichkeiten, dies zu tun. Hier sollen, wie auch in den folgenden Abschnitten, nur jeweils diejenigen aufgezeigt werden, die auch im vorliegenden System verwendet wurden. Die Vereinigung $\mu_{A \cup B}$ zweier Fuzzy-Mengen A und B mit den entsprechenden Zugehörigkeitsfunktionen μ_A und μ_B (ODER-Operator) wird definiert als:

$$\mu_{A \cup B}(x) := \max(\mu_A(x), \mu_B(x)) \quad . \quad (\text{B.3})$$

Dem UND-Operator entspricht die Schnittmenge $\mu_{A \cap B}$; sie ist definiert als:

$$\mu_{A \cap B}(x) := \min(\mu_A(x), \mu_B(x)) \quad . \quad (\text{B.4})$$

Abbildung B.2 veranschaulicht die UND- und ODER-Operatoren.

B.3 Fuzzy-Implikation und Fuzzy-Inferenz

Will man empirisches Systemwissen mit unscharfen Methoden modellieren, kann man auf die *Fuzzy-Implikation* zurückgreifen, um mit linguistischen Variablen Regeln in der Form

$$\text{WENN } x = A \quad \text{DANN } y = B \quad (\text{B.5})$$

aufzustellen. Für die natürlichsprachliche Regel „Wenn die Geschwindigkeit langsam ist, dann ist die Gefahr gering“ wäre die Eingangsgröße x die linguistische Variable „Geschwindigkeit“, die einen linguistischen Term „niedrig“ erfüllen muss, damit die Ausgangsgröße y (die Variable „Gefahr“) durch den Term „niedrig“ charakterisiert wird. Im Gegensatz zur klassischen Logik ist die Prämisse nicht entweder wahr oder falsch, sondern kann beliebige *Erfüllungsgrade* annehmen; entsprechend ist auch die Conclusio ein unscharfer Wert. Der Erfüllungsgrad einer Prämisse entspricht im einfachsten Fall dem Zugehörigkeitsgrad der entsprechenden Fuzzy-Menge.

Die Fuzzy-Implikation $\mu_{A \Rightarrow B}(x, y)$ definieren wir als:

$$\mu_{A \Rightarrow B}(x, y) := \min(\mu_A(x), \mu_B(y)) \quad . \quad (\text{B.6})$$

Diese Definition (die so genannte *Mamdani-Implikation*) hat insbesondere die wünschenswerte Eigenschaft, dass der Wahrheitsgehalt einer Schlussfolgerung nie höher sein kann als der der Prämisse. Anschaulich bedeutet dies, dass die Ausgangs-Fuzzy-Menge am Erfüllungsgrad der Eingangs-Menge „abgeschnitten“ wird. Die Auswertung eines Satzes von Regeln bezeichnet man als *Inferenz*. Eine grafische Zusammenstellung eines Inferenzvorganges zeigt Abbildung B.3.

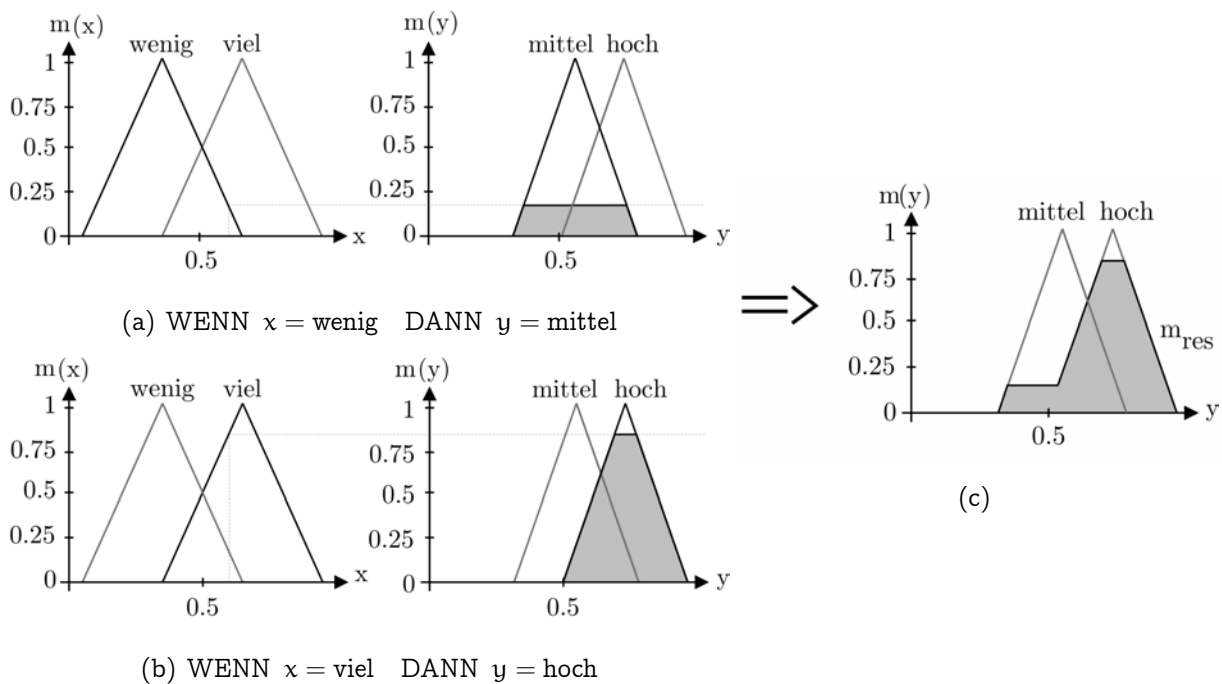


Abbildung B.3: Fuzzy-Inferenz mit zwei Regeln über einem Eingangsbereich X mit einem Ergebnisbereich Y . (a) (b) Die zugehörige Ergebnismenge wird auf der Höhe des jeweiligen Zugehörigkeitsgrades der Eingangsmenge abgeschnitten. (c) Die letztendlich resultierende Fuzzy-Menge ergibt sich durch Anwendung des Vereinigungsoperators auf die beiden Ergebnismengen.

Sind für eine Ausgangsgröße nach dem Inferenzvorgang mehrere Fuzzy-Mengen *aktiv* (d. h. der Zugehörigkeitsgrad des Eingangswerts ist größer als 0), werden diese mittels des ODER-Operators vereinigt. Die so entstandene Ergebnismenge μ_{res} muss nun noch *defuzzifiziert* werden, um einen scharfen Ausgabewert zu erhalten. Eine gebräuchliche Methode hierfür ist die so genannte *Schwerpunkt-Methode*, bei der sich der Ergebniswert aus der Projektion des Schwerpunktes der durch die Fuzzy-Menge beschriebenen Fläche auf die Achse des Wertebereichs ergibt. Der Schwerpunkt y_{res} errechnet sich dabei aus

$$y_{\text{res}} = \frac{\int y \mu_{\text{res}}(y) dy}{\int \mu_{\text{res}}(y) dy} \quad . \quad (\text{B.7})$$

Anhang C

Realisierte Fuzzy-Bewerter

Dieses Kapitel beschreibt die in dieser Arbeit realisierten Fuzzy-Bewerter anhand der definierten Fuzzy-Mengen und Regeln. Weitergehende Informationen zu den einzelnen Bewertungsvorgängen wie beispielsweise die Wahl der linguistischen Variablen sind den vorangehenden Kapiteln der Arbeit zu entnehmen. Entsprechende Verweise sind den einführenden Worten der folgenden Abschnitten zu entnehmen.

Die im Folgenden dargestellten Tabellen und Grafiken wurden automatisch aus den auf der Laufmaschine angewandten Regelsätzen generiert; daher auch die durchgängig englischen Bezeichner. Die Abbildung der Fuzzy-Mengen einer linguistischen Variable stellt neben dem Bezeichner der jeweiligen Fuzzy-Menge auch deren Einflussbreite (darunter in eckigen Klammern) sowie dessen Zentrum (oberhalb der Dreiecke) dar. An den Spitzen der Dreiecke ist der Zugehörigkeitsgrad zur Fuzzy-Menge gleich 1, auf der Grundlinie gleich 0. In diesem Zusammenhang ist noch zu bemerken, dass die Ränder bei trapezförmigen Fuzzy-Mengen nicht mehr korrekt skaliert dargestellt sind, da ansonsten oftmals die Übersichtlichkeit leiden würde. Die Regeln in den Tabellen sind von rechts nach links abzulesen. Ein „-“ steht dabei für „nicht beachtet“.

C.1 Odometrie – Bodenkontaktbewertung

Die Berechnung der Odometrie von Laufmaschinen nach Abschnitt 3.4 erfordert eine Gewichtung der Beine. In Abschnitt 3.4.2 wurde hierzu der Einsatz von Unschärfe Regeln vorgeschlagen. Während der Realisierungsphase auf LAURON IV wurden unterschiedliche Kombinationen von Fuzzy-Bewertern und eine Vielzahl an Regelsätzen untersucht (Abschnitt 7.1.2). Schließlich stellte sich die im Folgenden skizzierte Variante als die robusteste und im Zusammenhang mit der Odometrieberechnung am besten geeignete heraus. Abbildung C.1 führt die Fuzzy-Mengen der linguistischen Variablen auf. Tabelle C.1 bildet den vollständigen Regelsatz ab.

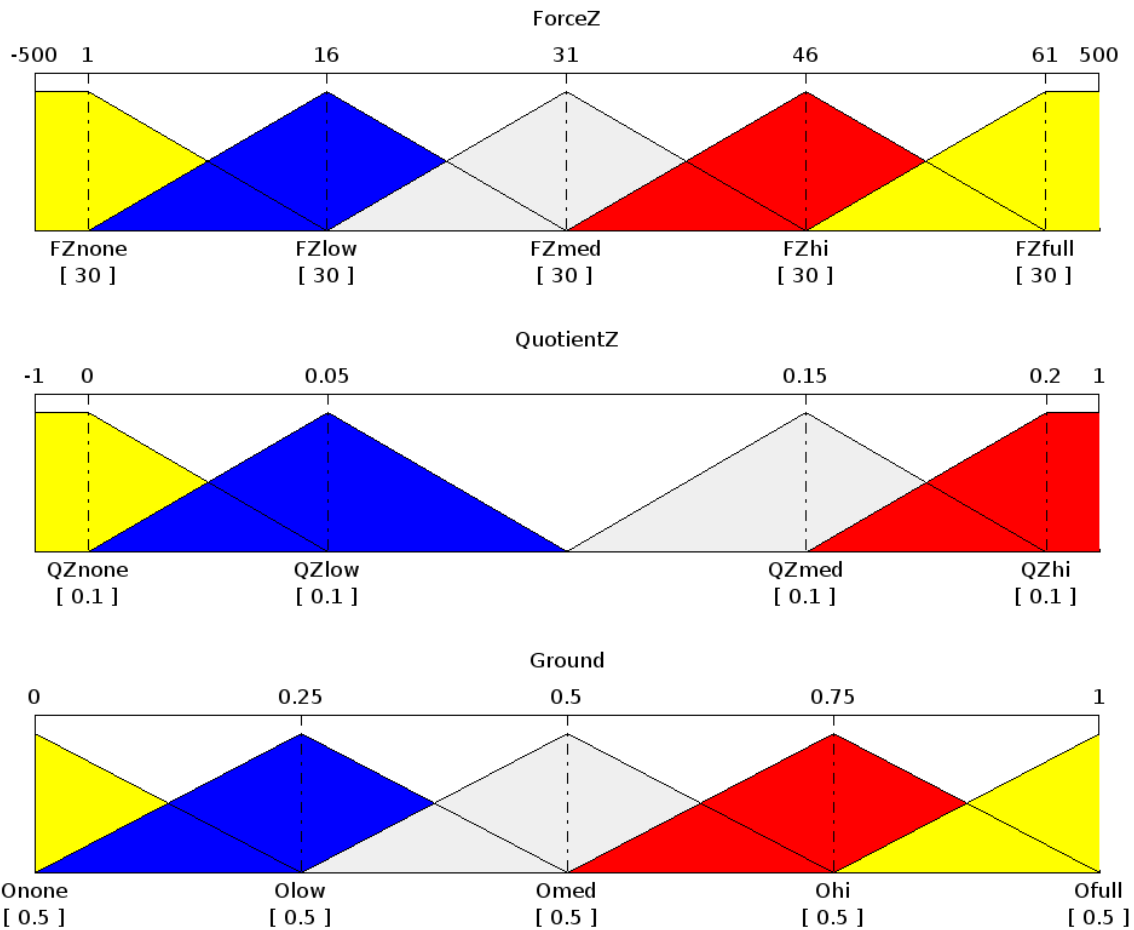


Abbildung C.1: Bodenkontaktbewertung: Definition der Fuzzy-Mengen der linguistischen Eingangsvariablen *ForceZ* (in z-Richtung wirkende Fußkraft), *QuotientZ* (relativer Anteil an der Summe über alle Füße $ForceZ / \sum_j ForceZ$) sowie der Ausgangsvariablen *Ground* (Intensität des Bodenkontaktes bzw. Gewichtung des Beines bei der Odometrieberechnung).

ForceZ	QuotientZ	Ground
FZnone	–	Onone
–	QZnone	Onone
FZlow	QZlow	Onone
FZlow	QZmed	Olow
FZlow	QZhi	Omed
FZmed	QZlow	Olow
FZmed	QZmed	Omed
FZmed	QZhi	Ohi
FZhi	QZlow	Omed
FZhi	QZmed	Ohi
FZhi	QZhi	Ofull
FZfull	–	Ofull

Tabelle C.1: Tabellarische Darstellung der Regeln zur Bewertung des Bodenkontakts *Ground*: Der *QuotientZ* dient zur Feinabstufung der auftretenden Kräfte *ForceZ*.

C.2 Umweltmodellierung – Einfügen neuer Sensordaten

Die Sensordatenfusion im 3D-Umweltmodell wird nach Abschnitt 4.7.3 durch einen Fuzzy-Bewerter unterstützt. Hierbei werden die Aufsetzpunkte der Fußsensoren sowie vor allem der am Kopf an LAURON IV montierte Abstandssensor verwendet (Abschnitt 7.1). Die Abbildungen C.2 und C.3 führen wiederum die Fuzzy-Mengen der linguistischen Variablen auf. Der vollständige Regelsatz ist in Tabelle C.2 verzeichnet.

C.3 Adaption an die Geländebedingungen

Um das Laufverhalten an die unmittelbar vorherrschenden Geländebedingungen anzupassen, werden nach Abschnitt 6.4 die Laufparameter des Roboters abhängig von den ermittelten Geländeeigenschaften adaptiert. Die Abbildungen C.4, C.5, C.6 und C.7 führen wiederum die Fuzzy-Mengen der linguistischen Variablen auf. Der vollständige Regelsatz ist in Tabelle C.3 verzeichnet.

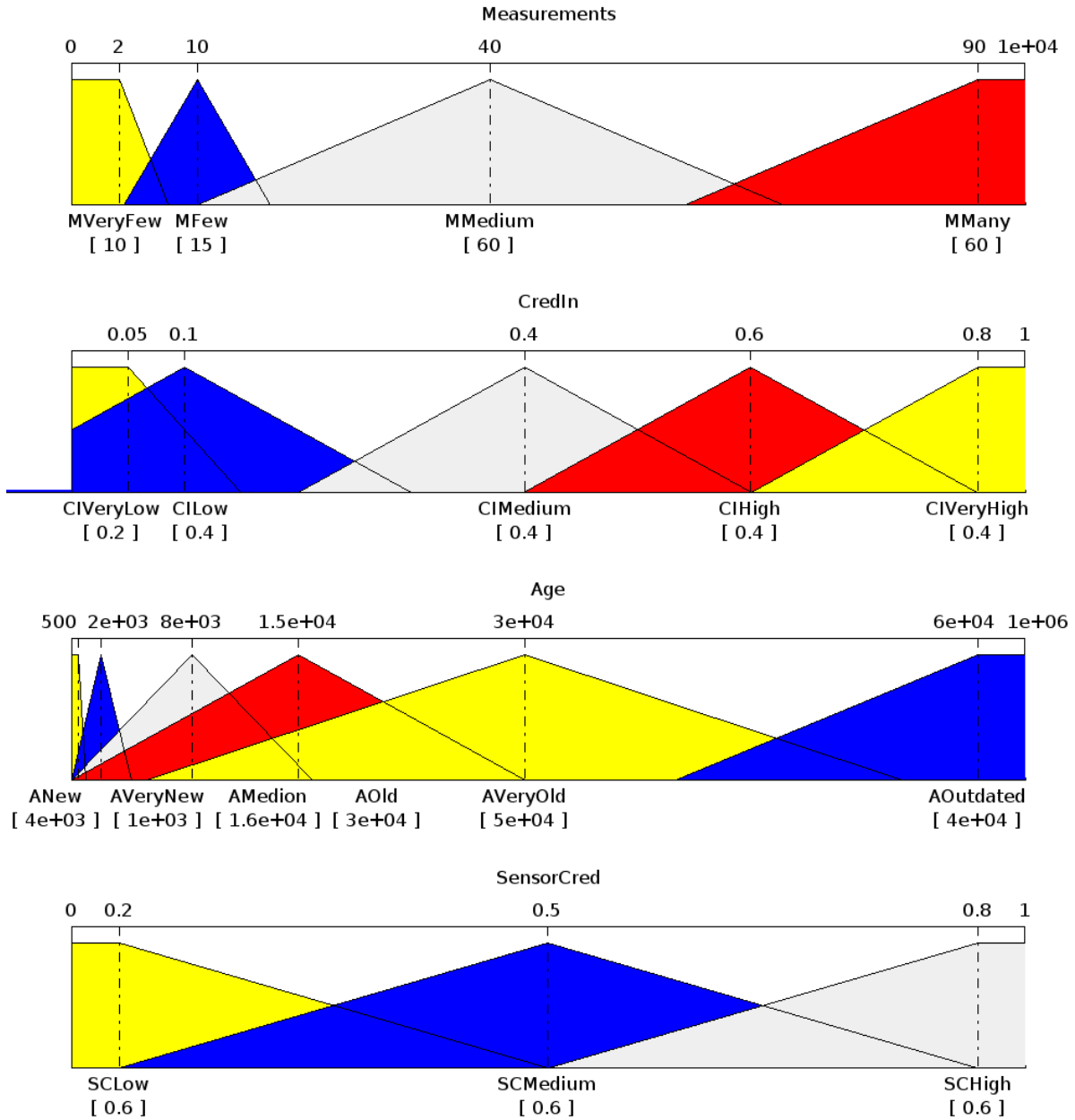


Abbildung C.2: Einfügen neuer Sensordaten in die dreidimensionale Urkarte: Definition der Fuzzy-Mengen der linguistischen Eingangsvariablen *Measurements* (Anzahl der bisherigen Messungen), *CredIn* (aktuell in der Gitterzelle gespeicherter Zuverlässigkeitswert), *Age* (Alter der letzten Messung) und *SensorCred* (Sensorqualität).

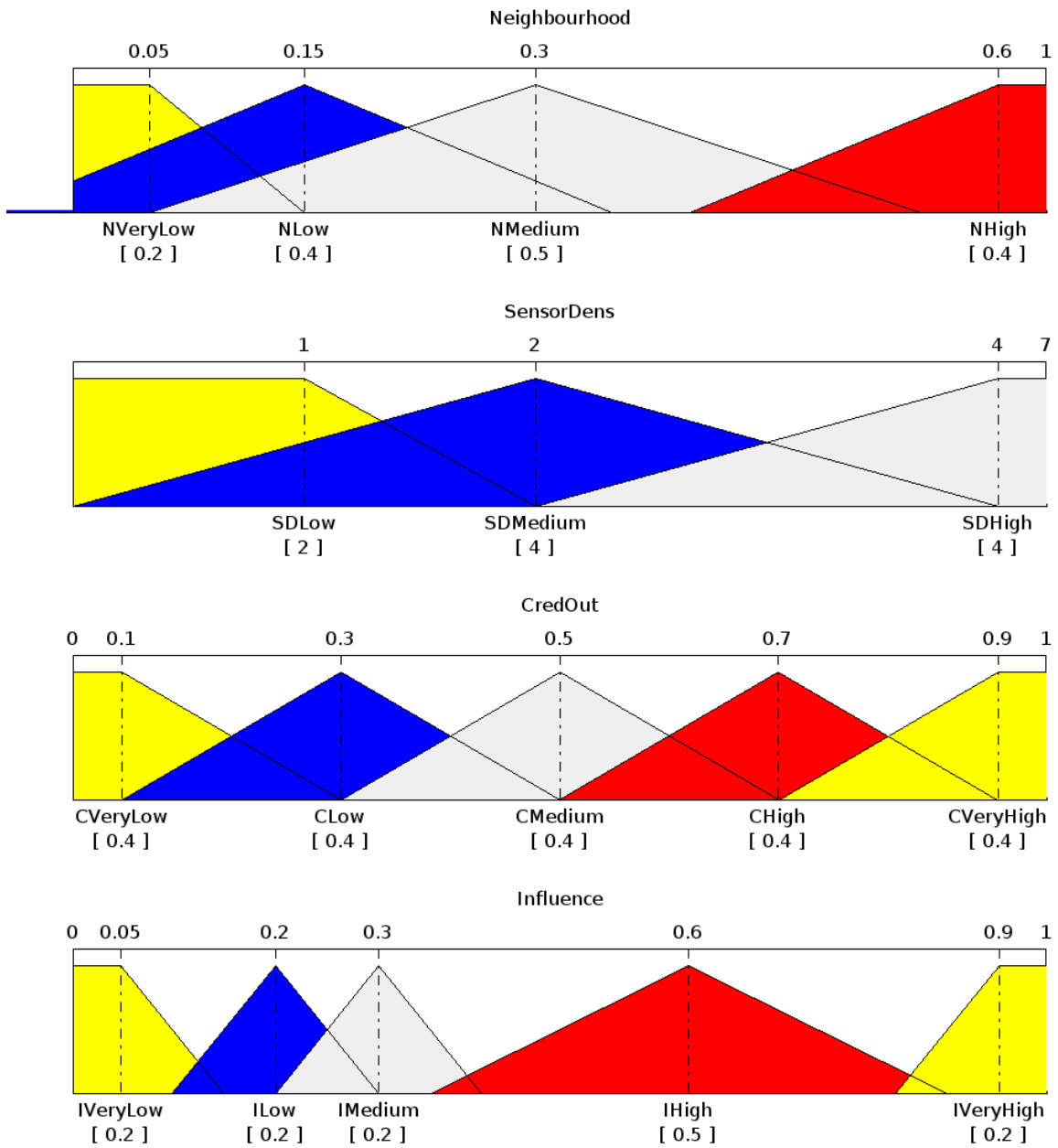


Abbildung C.3: Einfügen neuer Sensordaten in die dreidimensionale Urkarte: Definition der Fuzzy-Mengen der linguistischen Eingangsvariablen *Neighbourhood* (Nachbarschaftsbeziehung), *SensorDens* (Sensordichte bzw. Anzahl unterschiedlicher Datenquellen) sowie der Ausgangsvariablen *CredOut* (Zuverlässigkeitswert der Messung) und *Influence* (Einfluss der Messung)

Measurements	CredIn	Age	SensorCred	Neighbourhood	SensorDens	CredOut	Influence
MMany	-	-	-	-	-	CVeryHigh	IVeryLow
MMedium	-	-	-	-	-	CHigh	ILow
MFew	-	-	-	-	-	CMedium	IHigh
MVeryFew	-	-	-	-	-	CVeryLow	IVeryHigh
-	CVeryLow	-	-	-	-	-	IHigh
-	CILow	-	-	-	-	-	IMedium
-	CIMedium	-	-	-	-	-	ILow
-	CIHigh	-	-	-	-	-	IVeryLow
-	CIVeryHigh	-	-	-	-	-	IVeryLow
-	-	AVeryNew	-	-	-	-	IVeryLow
-	-	ANew	-	-	-	-	ILow
-	-	AOld	-	-	-	-	IMedium
-	-	AVeryOld	-	-	-	-	IHigh
-	-	AOutdated	-	-	-	-	IVeryHigh
-	-	-	SCLow	-	-	CLow	IVeryLow
-	-	-	SCMedium	-	-	CMedium	IMedium
-	-	-	SCHigh	-	-	CVeryHigh	IHigh
MMany	-	AVeryOld	-	-	-	-	IHigh
MMany	-	AOutdated	-	-	-	-	IVeryHigh
MMany	CILow	-	-	-	-	-	IVeryHigh
-	-	-	-	NVeryLow	-	CVeryLow	-
-	-	-	-	NLow	-	CLow	-
-	-	-	-	NMedium	-	CHigh	-
-	-	-	-	NHigh	-	CVeryHigh	-
-	-	-	-	-	SDLow	CMedium	-
-	-	-	-	-	SDMedium	CHigh	-
-	-	-	-	-	SDHigh	CVeryHigh	-

Tabelle C.2: Tabellarische Darstellung der Regeln zur Bewertung des Einflusses *Influence* einer neuen Messung sowie deren Zuverlässigkeit *CredOut*.

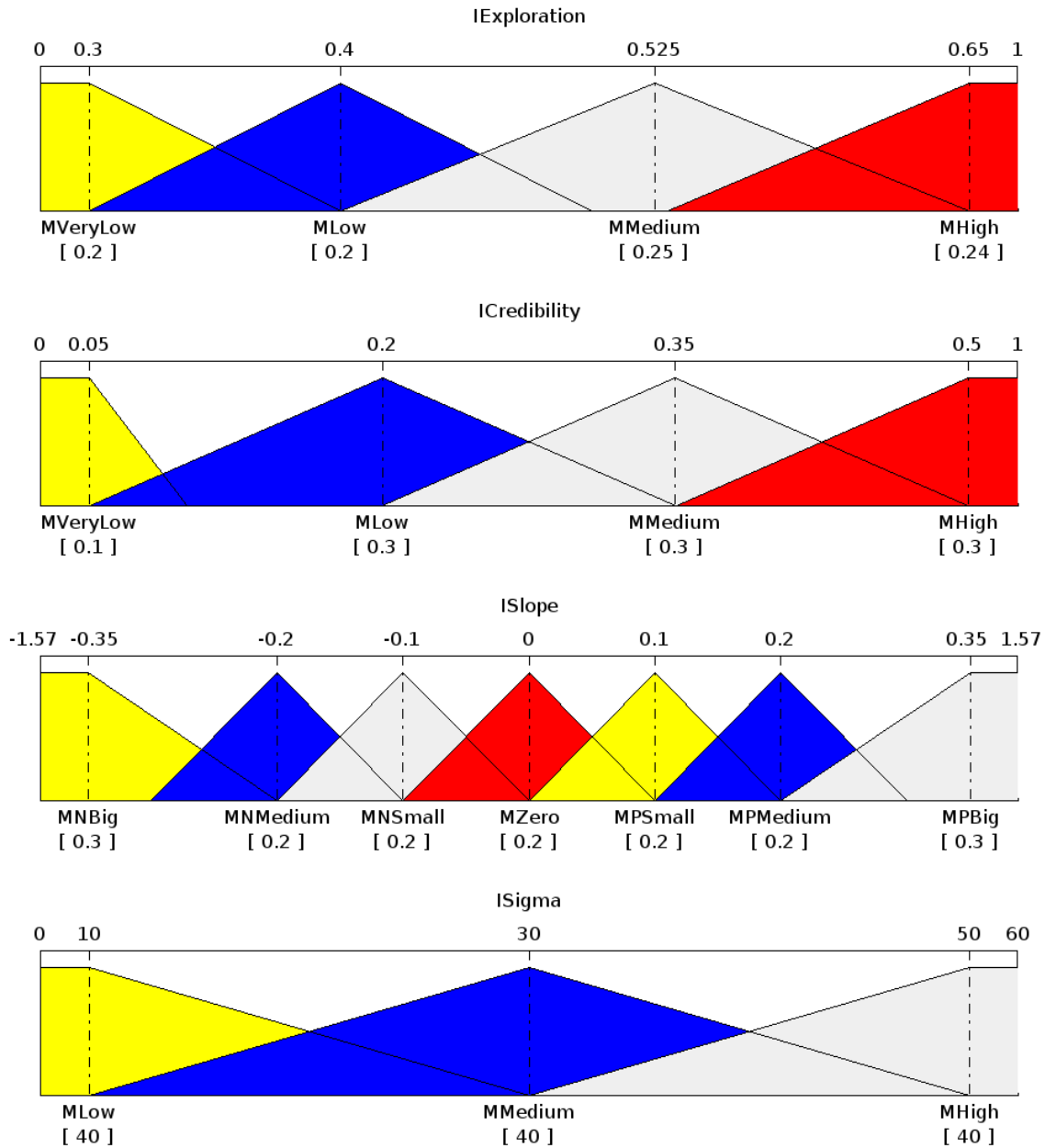


Abbildung C.4: Anpassung der Laufparameter: Definition der Fuzzy-Mengen der linguistischen Eingangsvariablen *Exploration* (Exploration), *Credibility* (Zuverlässigkeit), *Slope* (Steigung), sowie *Sigma* (Standardabweichung)

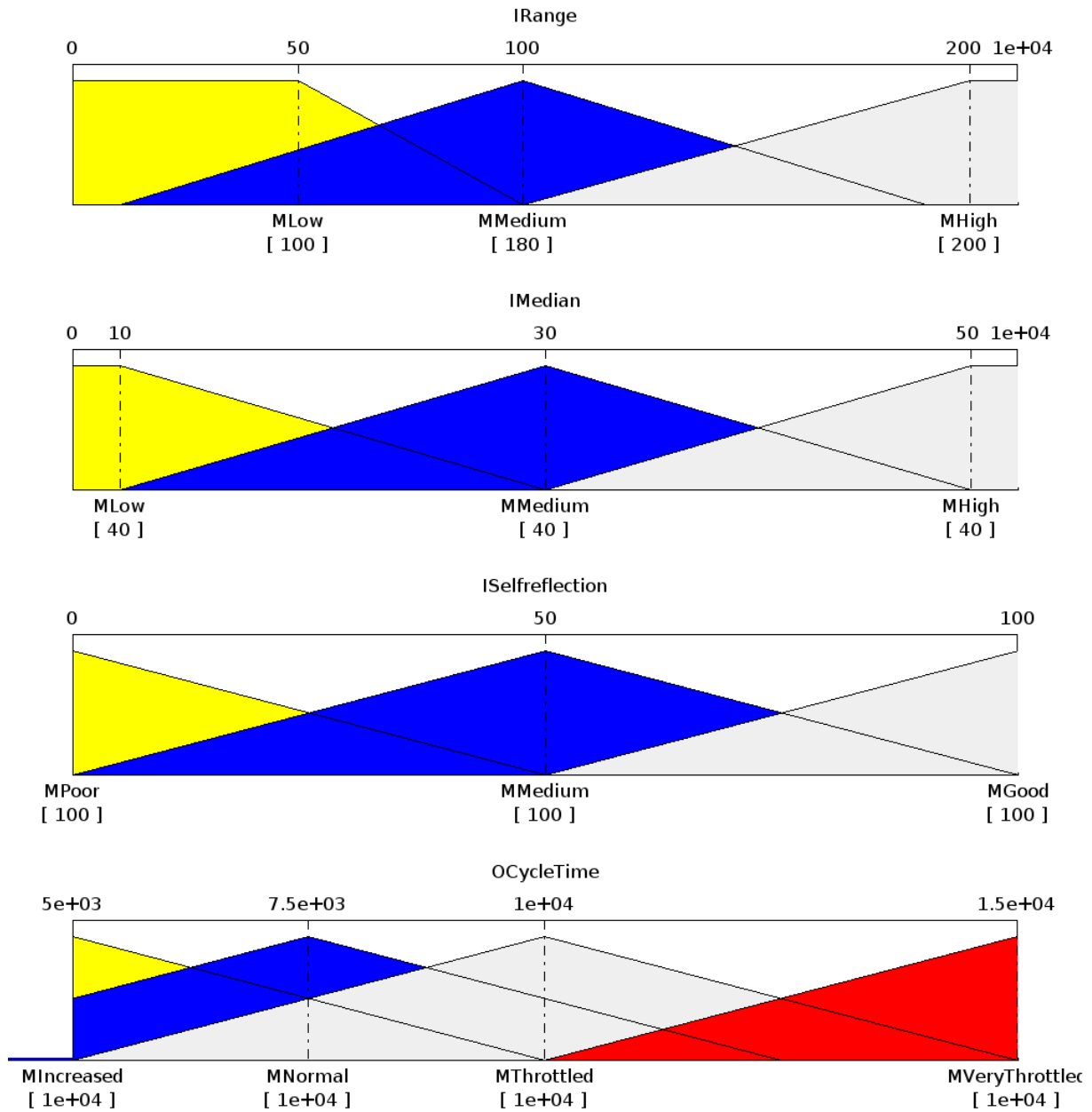


Abbildung C.5: Anpassung der Laufparameter: Definition der Fuzzy-Mengen der linguistischen Eingangsvariablen *Range* (Spannweite), *Median* (Median), *Selfreflection* (Selbstreflexion Verhaltenssteuerung), sowie der Ausgangsvariablen *CycleTime* (Zykluszeit)

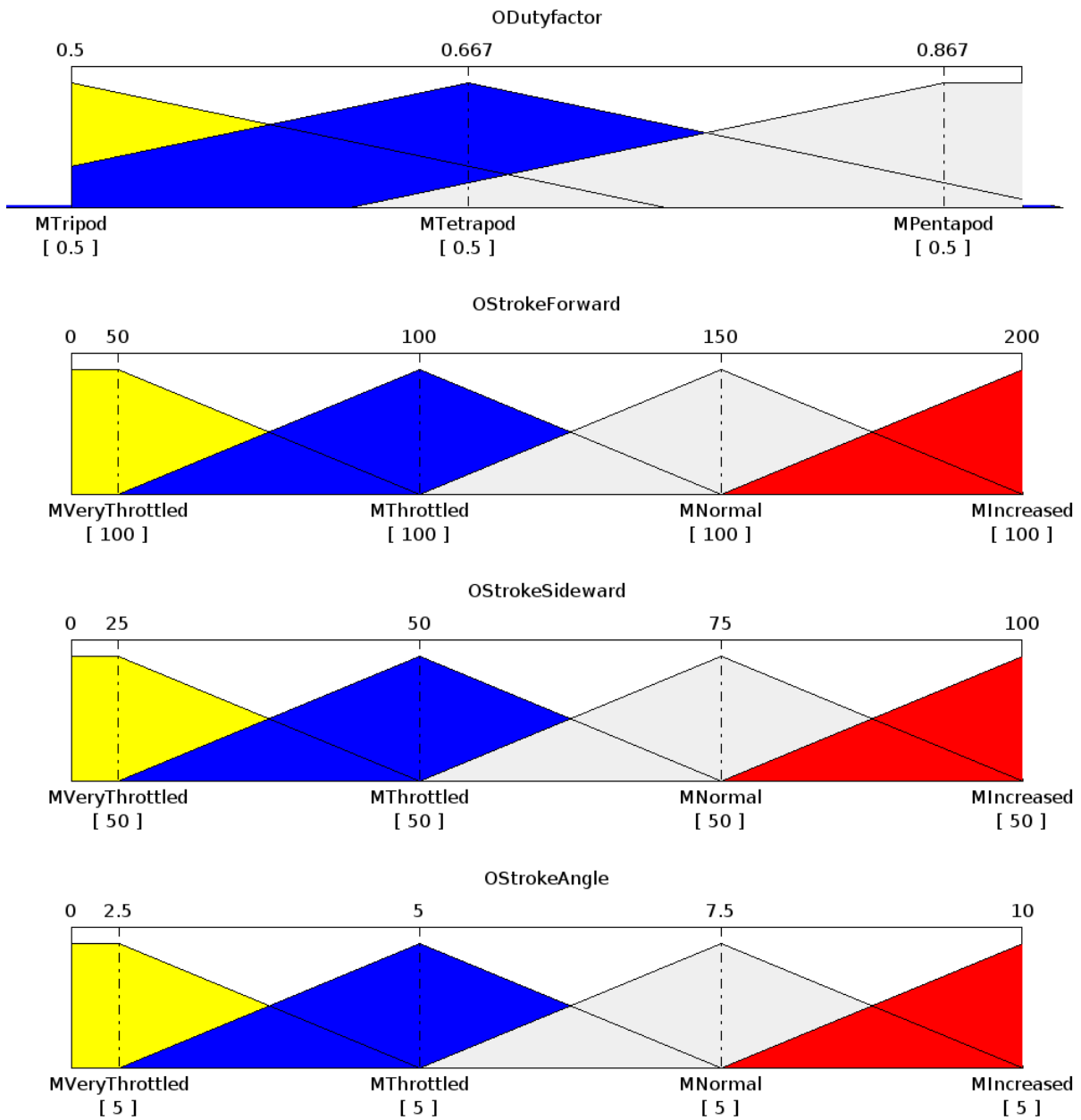


Abbildung C.6: Anpassung der Laufparameter: Definition der Fuzzy-Mengen der linguistischen Ausgangsvariablen *Dutyfactor* (Belastungsfaktor), *StrokeForward* (Beinschlag nach vorne), *StrokeSideward* (Beinschlag zur Seite), sowie *StrokeAngle* (Beinschlag um die Kurve).

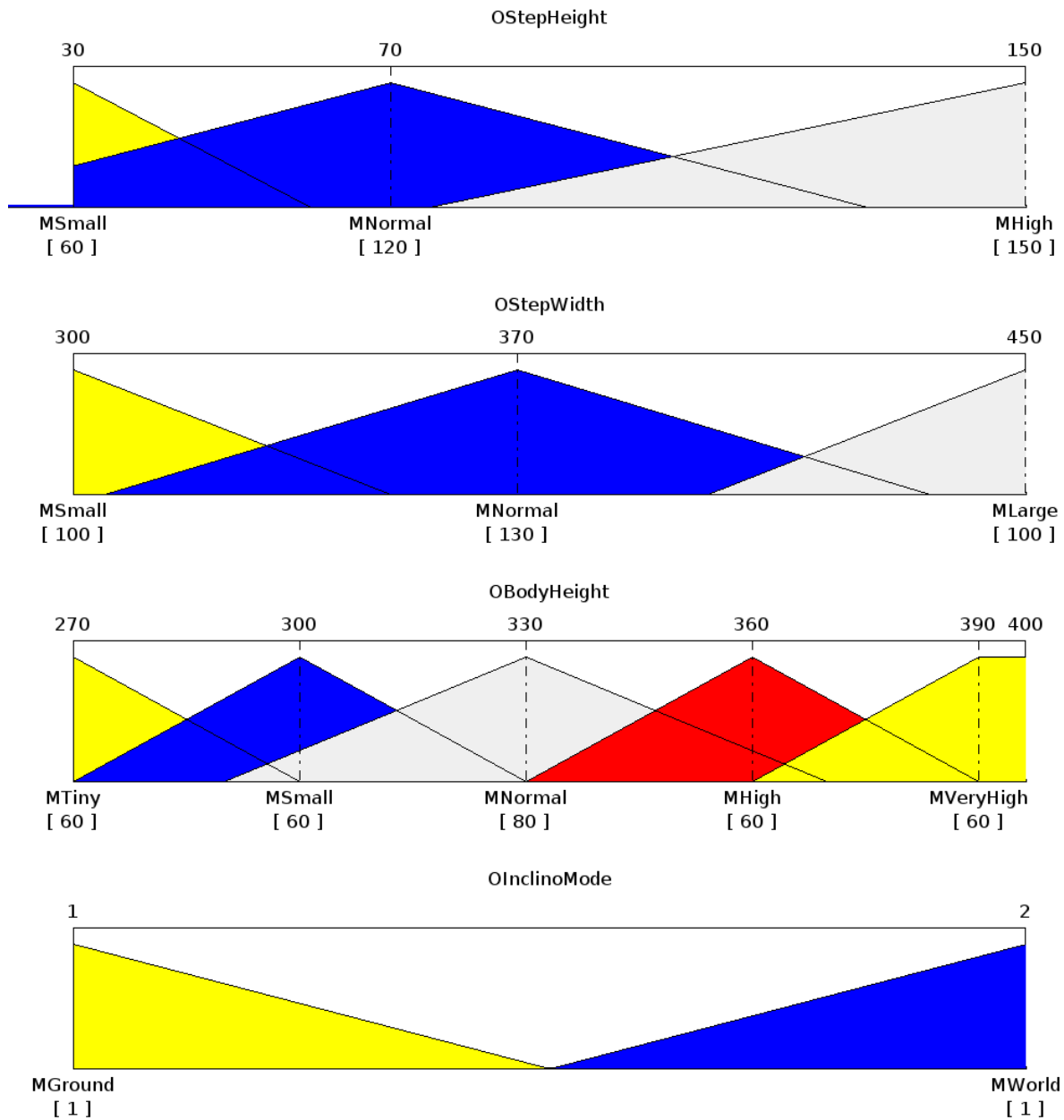


Abbildung C.7: Anpassung der Laufparameter: Definition der Fuzzy-Mengen der linguistischen Ausgangsvariablen *StepHeight* (Schritthöhe), *StepWidth* (Schrittbreite), *BodyHeight* (Körperhöhe), sowie *InclinoMode* (Körperneigungsart).

IEExploration	ICredibility	ISlope	ISigma	IRange	IMedian	ISelfreflection	OCycleTime	ODutyfactor	OStrokeForward	OStrokeSideward	OStrokeAngle	OStepHeight	OStepWidth	OBodyHeight	OInclinoMode
MHigh	-	-	-	-	-	-	MNormal	-	MNormal	MNormal	MNormal	-	-	-	-
MMedium	-	-	-	-	-	-	MThrottled	-	MNormal	MNormal	MNormal	-	-	-	-
MLow	-	-	-	-	-	-	MVeryThrottled	-	MThrottled	MThrottled	MThrottled	-	-	-	-
MVeryLow	-	-	-	-	-	-	MVeryThrottled	-	MThrottled	MThrottled	MThrottled	-	-	-	-
-	MVeryLow	-	-	-	-	-	MVeryThrottled	-	MVeryThrottled	MVeryThrottled	MVeryThrottled	MHigh	-	-	-
-	MLow	-	-	-	-	-	MThrottled	-	MThrottled	MThrottled	MThrottled	MHigh	-	-	-
-	MMedium	-	-	-	-	-	MNormal	-	MNormal	MNormal	MNormal	MNormal	-	-	-
-	MHigh	-	-	-	-	-	MNormal	-	MNormal	MNormal	MNormal	MNormal	-	-	-
-	-	MNBig	-	-	-	-	MVeryThrottled	MPentapod	MThrottled	MThrottled	MThrottled	MHigh	MLarge	MTiny	MGround
-	-	MPBig	-	-	-	-	MVeryThrottled	MThrottled	MThrottled	MThrottled	MThrottled	MHigh	MLarge	MTiny	MGround
-	-	MNMedium	-	-	-	-	MThrottled	MThrottled	MNormal	MNormal	MNormal	MHigh	MNormal	MSmall	MGround
-	-	MPMedium	-	-	-	-	MThrottled	MThrottled	MNormal	MNormal	MNormal	MHigh	MNormal	MNormal	MGround
-	-	MNSmall	-	-	-	-	MNormal	MThippod	MNormal	MNormal	MNormal	MNormal	MNormal	MNormal	MWorld
-	-	MPSmall	-	-	-	-	MNormal	MThippod	MNormal	MNormal	MNormal	MNormal	MNormal	MNormal	MWorld
-	-	MZero	-	-	-	-	MNormal	MThippod	MIncreased	MIncreased	MIncreased	MNormal	MNormal	MNormal	MWorld
-	-	-	MLow	-	-	-	MNormal	-	MNormal	MNormal	MNormal	MNormal	-	-	-
-	-	-	MMedium	-	-	-	MThrottled	-	MThrottled	MThrottled	MThrottled	MHigh	-	-	-
-	-	-	MHigh	-	-	-	MVeryThrottled	MThrottled	MThrottled	MThrottled	MThrottled	MHigh	-	-	-
-	-	-	-	MLow	-	-	MIncreased	-	MIncreased	MIncreased	MIncreased	MSmall	-	-	-
-	-	-	-	MMedium	-	-	MNormal	-	MNormal	MNormal	MNormal	MNormal	-	-	-
-	-	-	-	MHigh	-	-	MThrottled	-	MThrottled	MThrottled	MVeryThrottled	MNormal	-	-	-
-	-	-	-	-	MLow	-	MNormal	-	MNormal	MNormal	MNormal	MNormal	-	-	-
-	-	-	-	-	MMedium	-	MThrottled	-	MThrottled	MThrottled	MThrottled	MNormal	-	-	-
-	-	-	-	-	MHigh	-	MVeryThrottled	-	MThrottled	MThrottled	MThrottled	MHigh	-	-	-
-	-	-	-	-	-	MPoor	MVeryThrottled	MPentapod	MVeryThrottled	MVeryThrottled	MVeryThrottled	MHigh	-	-	-
-	-	-	-	-	-	MMedium	MThrottled	MThrottled	MThrottled	MThrottled	MThrottled	MNormal	-	-	-
-	-	-	-	-	-	MGood	MNormal	MThippod	MNormal	MNormal	MNormal	MNormal	-	-	-

Tabelle C.3: Tabellarische Darstellung der Regeln zur Anpassung des Laufverhaltens von LAURON.

Symbolverzeichnis

Formatierungskonventionen mathematischer Symbole

- a, b, c Variablen
- $\mathbf{a}, \mathbf{b}, \mathbf{c}$ Vektoren
- A, B, C Matrizen

Symbole zur Lagebeschreibung

- \mathbf{p} Positionsvektor
- x, y, z Komponenten des Positionsvektors \mathbf{p}
- \mathbf{v} Vektor der Geschwindigkeiten
- $\boldsymbol{\psi}$ Orientierungsvektor
- α, β, γ Komponenten des Orientierungsvektors $\boldsymbol{\psi}$
- $\boldsymbol{\omega}$ Vektor der Winkelgeschwindigkeiten
- R_r^g Rotationsmatrix für Komponentenübergang von r nach g und gleichzeitig Rotationsmatrix der Roboterorientierung in g
- \mathbf{p}_r^g Translationsvektor für Komponentenübergang von r nach g und gleichzeitig Roboterposition in g

Symbole bei der Verwendung im Kalman-Filter

- \mathbf{u} optionale Steuereingabe
- \mathbf{s} Vektor des Systemrauschens
- \mathbf{m} Vektor des Messrauschens
- \mathbf{x} Zustandsvektor
- $\Delta\mathbf{x}$ Fehlerzustandsvektor
- \mathbf{z} Messvektor
- E Matrix zur Einbringung der optionalen Steuereingabe
- F Matrix des Systemmodells
- G Matrix zur Transformation des Systemrauschens
- H Matrix, die Messung und Zustand in Verbindung setzt
- K Matrix des Kalman-Faktors
- P Kovarianzmatrix des geschätzten Zustands
- M Kovarianzmatrix des Messrauschens
- N Kovarianzmatrix des Systemrauschens
- Φ Matrix des approximierten diskreten Systemmodells

Symbole bei der Verwendung im Bahnplaner

Mz	Körpermittelpunktzelle
$\mathbf{p} = (Mz_x, Mz_y, Mz_\gamma)$	diskrete Roboterposition
c	Gitterzelle
Mz_x, Mz_y	diskrete Zellindizes der Körpermittelpunktzelle
$Mz_\gamma \in \{\text{Nord, West, Süd, Ost}\}$	diskrete Orientierung des Roboters
L_{Kpl}	Länge des Zentralkörpers
L_{Kpb}	Breite des Zentralkörpers
L_{Kph}	Höhe des Zentralkörperunterteils
Sh_i	Gitterzelle der Schulter i
L_{Shb}	seitlicher Abstand der Schulterzellen
L_{Shl}	Abstand der Schulterzellen in Längsrichtung
Ar_i	Beinarbeitsbereich des Beines i
L_{Arl}	Länge eines Arbeitsbereiches
L_{Arb}	Breite eines Arbeitsbereiches
L_{Ardl}	Abstand der Arbeitsbereiche in Längsrichtung
L_{Ardb}	seitlicher Abstand der Arbeitsbereiche
Os	Oberschenkel
Us	Unterschenkel
L_{Ush}	Höhe bzw. Länge der Unterschenkel
L_{Smin}	minimale Stemmhöhe der Beine
L_{Smax}	maximale Stemmhöhe der Beine
L_{Kpn}	Neigung des Zentralkörpers um die y -Achse
Fp	Fußaufsetzpunkt
d_{Fp}^{Sh}	Abstand zwischen Schulter und Fußaufsetzpunkt
m_{Os}	maximale Steigung des Oberschenkels
h_{Fp}^{Sh}	minimale Schulterhöhe über dem Fußaufsetzpunkt
h_a^{Sh}	minimale absolute Höhe der Schulter
h_{max}^{Kp}	maximale Höhe des Zentralkörpers
\overline{Sh}_j	Schulterachse j
$h_{max}^{Sh_j}$	maximale Höhe der Schulterachse j
$h_{min}^{Sh_j}$	minimale Höhe der Schulterachse j
ZF	Zellfläche
c_{ZF}	Repräsentant einer Zellfläche
A_p	Stellungsmachbarkeit
A_t	Geländezugänglichkeit
\overline{A}_t	Geländekomplexität
\widetilde{A}_t	3D-Geländekomplexität
F	Freibereich der Laufmaschine
R_F, R_{Kp}, R_{Ar}	maximaler Anteil unbekannter Zellen im Freibereich, unterhalb des Zentralkörpers und im Arbeitsbereich
e	Explorationswert für unbekannte Bereiche

Symbole mit besonderer Bedeutung

$\ \dots\ $	euklidische Norm
t	Zeitvariable
1	Einheitsmatrix
$k-1, k, k+1$	Indizes diskreter Zeitschritte

Abbildungsverzeichnis

1.1	Aufgabenszenario für die Navigation	4
2.1	Sechsbeiniger Roboter ADAPTIVE SUSPENSION VEHICLE	10
2.2	Planetenerkundungsroboter AMBLER	11
2.3	DANTE II – Ein Roboter zur Vulkanerkundung.	12
2.4	Die vierbeinige Forschungsplattform JROB-2.	13
2.5	Dynamische sechsbeinige Laufmaschine RHEX	14
2.6	Achtbeinige Laufmaschine SCORPION	15
2.7	Zentaurenartiger Serviceroboter WORKPARTNER	16
2.8	Mars Rover ROCKY 7	17
3.1	Koppelnavigation	25
3.2	Struktur eines Strapdown Inertial-Navigationssystems.	27
3.3	Das Navstar-GPS-System	31
3.4	Ablauf der Odometrieberechnung von Laufmaschinen	42
3.5	Typische Anwendung des Kalman-Filters	43
3.6	Ablauf des Kalman-Filters	46
3.7	Direktes Kalman-Filter	47
3.8	Indirektes Kalman-Filter ohne Rückkopplung	48
3.9	Indirektes Kalman-Filter mit Rückkopplung	49
3.10	Das realisierte Kalman-Filter	50
3.11	Ablauf des Lokalisationsalgorithmus	56
4.1	Beispiel einer geometrischen und einer topologischen Karte	65
4.2	Verschieben der 3D-Umgebung und Erzeugung einer 2½D-Karte	72
4.3	Schichtweiser Aufbau der Karten des Umweltmodells	73
4.4	Interaktion von Steuerungsebene und verschiedener Karten	74
4.5	Repräsentation einer Unterteilung einer Fläche in einem Quadtree	75
4.6	Speicherbedarf einer Szene in einem Quadtree	76
4.7	Datenfluss bei der Bewertung der Gitterzelle	81
4.8	Fuzzy-Mengen für die Eingangsgröße „Anzahl der Messungen“	82
4.9	Verlauf von Einflussgrößen bei unterschiedlichen Eingabewerten	83
5.1	Planungsverfahren: Dekomposition	91
5.2	Planungsverfahren: evolutionäre Algorithmen	93
5.3	Potenzialfeldverfahren	94

5.4	Laufmaschinenmodell: Basismodell nach Bai und Low	98
5.5	Zentralkörpermodell: Draufsicht mit Beinarbeitsbereichen	102
5.6	Zentralkörpermodell: Front- und Seitenansicht	103
5.7	Beinmodell: Draufsicht und Frontansicht	104
5.8	Bewertung: Minimale und maximale Höhe einer Schulterachse	106
5.9	Bewertung: Gültige Stellung der Schulterachsen	108
5.10	Bewertung: Kombinationsvorgang mit vier Beinarbeitsbereichen Ar_i . .	110
5.11	Bewertung: Freibereich der Laufmaschine	111
6.1	Orientierungslaufkarten	118
6.2	Geländeklassifikation eines schnell fahrenden Geländewagens	120
6.3	Finden eines geeigneten Fußaufsetzpunktes	126
7.1	Die LAURON-Ahnengalerie	130
7.2	Sensorik zur Umwelterfassung von LAURON IV	132
7.3	Bodenkontaktbewertung der Fußsensordaten	134
7.4	Kalibrierung der Magnetfeldsensoren	135
7.5	Stabilisierung der Magnetfeldsensoren	136
7.6	Einfluss des Laufens auf die Magnetfeldsensoren	137
7.7	Lokalisation: Odometrieexperimente	141
7.8	Lokalisation: Experimente zur Orientierungskorrektur	143
7.9	Lokalisation: Laufen im Freien	144
7.10	Lokalisation: Laufen im Kies	145
7.11	Lokalisation: Höhendaten beim Laufen im Freien	146
7.12	Virtueller Sensor zum Eintragen von freien Bereichen	149
7.13	Aufbau des Kartierungsversuches	150
7.14	Kartierungsexperiment: Schrittweiser Aufbau der 3D Urkarte A	151
7.15	Kartierungsexperiment: Schrittweiser Aufbau der 3D Urkarte B	152
7.16	Kartierungsexperiment: Die globale $2\frac{1}{2}D$ -Karte	153
7.17	Bahnplanung: Aufbau Grabenexperiment	155
7.18	Bahnplanung: Bewertung der Grabenexperimente A	156
7.19	Bahnplanung: Bewertung der Grabenexperimente B	157
7.20	Bahnplanung: Aufbau Säulenexperiment	158
7.21	Bahnplanung: Säulenexperiment mit Einflussgrenze	158
7.22	Bahnplanung: Säulenexperiment ohne Einflussgrenze	159
7.23	Bahnplanung: Aufbau des Bahnplanungsexperiments	160
7.24	Bahnplanung: Vollständig bekanntes Umweltmodell	161
7.25	Bahnplanung: Unvollständiges Umweltmodell	162
7.26	Adaption: Eingangsgrößen der Laufparameter-Bewertung	164
7.27	Adaption: Ausgangsgrößen der Laufparameter-Bewertung	165
7.28	Waldszenario: Foto	168
7.29	Waldszenario: Aufbau der dreidimensionalen Urkarte	169
7.30	Waldszenario: $2\frac{1}{2}D$ -Karte und geplanter Weg	170
7.31	Waldszenario: Lokalisation	171

7.32	Waldszenario: Verifikation des Plans	172
A.1	Koordinatensysteme	180
A.2	Koordinatensysteme Lauron	181
B.1	Beispiel für Fuzzy-Mengen	188
B.2	UND- und ODER-Operator auf Fuzzy-Mengen	189
B.3	Beispiel einer Fuzzy-Inferenz	190
C.1	Fuzzy-Mengen zur Bodenkontaktbewertung	192
C.2	Fuzzy-Mengen zum Einfügen neuer Sensordaten ins Umweltmodell: Teil 1	194
C.3	Fuzzy-Mengen zum Einfügen neuer Sensordaten ins Umweltmodell: Teil 2	195
C.4	Fuzzy-Mengen zur Anpassung der Laufparameter: Teil 1	197
C.5	Fuzzy-Mengen zur Anpassung der Laufparameter: Teil 2	198
C.6	Fuzzy-Mengen zur Anpassung der Laufparameter: Teil 3	199
C.7	Fuzzy-Mengen zur Anpassung der Laufparameter: Teil 4	200

Tabellenverzeichnis

1.1	Aufbau der Arbeit	6
2.1	Autonomiegrade von DANTE II	12
3.1	Übersicht zu erwartender GPS-Genauigkeiten	32
3.2	Beurteilung der Lokalisationsverfahren	36
4.1	Gegenüberstellung der Eigenschaften geometrischer und topologischer Karten	67
4.2	Beurteilung der Umweltmodellierungsverfahren	70
5.1	Beurteilung der Planungsverfahren	95
5.2	Abschließender Vergleich der Bahnplaner für Laufmaschinen	115
C.1	Fuzzy-Regeln zur Bodenkontaktbewertung	193
C.2	Fuzzy-Regeln zum Einfügen neuer Sensordaten ins Umweltmodell	196
C.3	Fuzzy-Regeln zur Anpassung der Laufparameter	201

Literaturverzeichnis

- [Albiez et al. 2005] ALBIEZ, J. ; GIESLER, B. ; LELLMANN, J. ; ZÖLLNER, J.M. ; DILLMANN, R.: Virtual immersion for tele-controlling a hexapod robot. In: *Proceedings of the 8th International Conference on Climbing and Walking Robots*, 2005, S. 1087–1094
- [Andrade et al. 1998] ANDRADE, G. ; AMAR, F. B. ; BIDAUD, P. ; CHATILA, R.: Modeling Robot-Soil Interaction for Planetary Rover Motion Control. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998, S. 576–581
- [Arbuckle et al. 2002] ARBUCKLE, D. ; HOWARD, A. ; MATARIĆ, M. J.: Temporal Occupancy Grids: a Method for Classifying the Spatio-Temporal Properties of the Environment. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, S. 409–414
- [Arleo et al. 1999] ARLEO, A. ; R. MILLÁN, J. del ; FLOREANO, D.: Efficient Learning of Variable-Resolution Cognitive Maps for Autonomous Indoor Navigation. In: *IEEE Transactions on Robotics and Automation* 15 (1999), Nr. 6
- [Asa et al. 2004] ASA, K. ; ISHIMURA, K. ; WADA, M.: Adaptive behavior to environment of a humanoid robot with CPG. In: *SICE Annual Conference*, 2004, S. 208–213
- [Bai und Low 2001] BAI, S. ; LOW, K. H.: Body Trajectory Generation for Legged Locomotion Systems Using a Terrain Evaluation Approach. In: *IEEE International Conference on Robotics and Automation*, 2001, S. 2279–2284
- [Bai et al. 2002] BAI, S. ; LOW, K. H. ; TEO, M. Y.: Path Generation of Walking Machines in 3D Terrain. In: *IEEE International Conference on Robotics and Automation*, 2002, S. 2216–2221
- [Bär 2005] BÄR, Tobias: *Modellgestützte Adaption von Laufverhalten im Gelände*, Forschungszentrum Informatik (FZI), Studienarbeit, Mai 2005
- [Bares und Wettergreen 1999] BARES, J. E. ; WETTERGREEN, D. S.: Dante II: Technical Description, Results, and Lessons Learned. In: *The International Journal of Robotics Research* 18 (1999), Nr. 7, S. 621–649

- [Barshan und Durrant-Whyte 1995] BARSHAN, B. ; DURRANT-WHYTE, H. F.: Inertial Navigation Systems for Mobile Robots. In: *IEEE Transactions on Robotics and Automation* 11 (1995), Nr. 3, S. 328–342
- [Bellutta et al. 2000] BELLUTTA, P. ; MANDUCHI, R. ; MATTHIES, L. ; OWENS, K. ; RANKIN, A.: Terrain Perception for DEMO III. In: *IEEE Intelligent Vehicles Symposium*, 2000, S. 326–331
- [Biewer 1997] BIEWER, B.: *Fuzzy-Methoden*. Springer-Verlag Berlin Heidelberg, 1997
- [Bihari et al. 1989] BIHARI, T. E. ; WALLISER, T. M. ; PATTERSON, M. R.: Controlling the Adaptive Suspension Vehicle. In: *Computer* 22 (1989), Nr. 6, S. 59–65
- [Borenstein et al. 1996] BORENSTEIN, J. ; EVERETT, H. R. ; FENG, L.: "Where am I?" Sensors and Methodes for Mobile Robot Positioning / The University of Michigan. 1996. – Forschungsbericht.
- [Borenstein und Feng 1994] BORENSTEIN, J. ; FENG, L.: UMBmark – A method for measuring, comparing and correcting dead-reckoning errors in mobile robots / University of Michigan. 1994. – Forschungsbericht.
- [Borenstein und Koren 1991] BORENSTEIN, J. ; KOREN, Y.: The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots. In: *IEEE Transactions on Robotics and Automation* 7 (1991), Nr. 3, S. 278–288
- [Brockhaus 1984] BROCKHAUS, F. A. (Hrsg.): *Der Große Brockhaus*. Wiesbaden, 1984
- [Bronstein und Semendjajew 1991] BRONSTEIN, I. N. ; SEMENDJAJEW, K. A.: *Taschenbuch der Mathematik*. Teubner Verlag, 1991
- [Celaya und Porta 1998] CELAYA, E. ; PORTA, J. M.: A Control Structure for the Locomotion of a Legged Robot on Difficult Terrain. In: *IEEE Robotics and Automation Magazine* 5 (1998), Nr. 2, S. 43–51
- [Chen und Kumar 1996] CHEN, C.-H. ; KUMAR, V.: Motion Planning of Walking Robots in Environment with Uncertainty. In: *IEEE International Conference on Robotics and Automation*, 1996, S. 3277–3282
- [Cheng et al. 2005] CHENG, Y. ; MAIMONE, M. ; MATTIES, L.: Visual Odometry on the Mars Exploration Rovers. In: *IEEE Conference on Systems, Man and Cybernetics*, 2005, S. 903–910
- [Chestnutt et al. 2005] CHESTNUTT, J. ; LAU, M. ; CHEUNG, G. ; KUFFNER, J. ; HODGINS, J. ; KANADE, T.: Footstep Planning for the Honda ASIMO Humanoid. In: *IEEE International Conference on Robotics and Automation*, 2005, S. 631–636

- [Chin et al. 2001] CHIN, Y. T. ; WANG, H. ; TAY, L. P. ; SOH, Y. C.: Vision Guided AGV using Distance Transform. In: *32nd International Symposium on Robotics*, 2001, S. 1416–1421
- [Cho et al. 2001] CHO, H.-K. ; CHO, Y.-J. ; YOU, B.-J.: Integration of Schema-Based Behaviours and Variable-Resolution Cognitive Maps for Stable Indoor Navigation. In: *IEEE International Conference on Robotics and Automation*, 2001, S. 3618–3623
- [Coombs et al. 2000] COOMBS, D. ; MURPHY, K. ; LACAZE, A. ; LEGOWIK, S.: Driving Autonomously Offroad up to 35 km/h. In: *IEEE Intelligent Vehicles Conference*, 2000, S. 186–191
- [Cordes und Bührle 1996] CORDES, S. ; BÜHRLE, P.: LAURON II- Simulation und Realisierung einer sechsbeinigen Laufmaschine. In: *Autonome Mobile Systeme*, 1996, S. 190–203
- [Cordes 1992] CORDES, Stefan: *Das INSECT-Modell: Entwicklung und Konstruktion eines Modells einer 6-beinigen Laufmaschine zur Generierung von Beintrajektorien*, Forschungszentrum Informatik (FZI), Studie, April 1992
- [Cordes 1994] CORDES, Stefan: *Konzeption und Realisierung einer flexiblen Steuerungsarchitektur für eine sechsbeinige Laufmaschine*, Forschungszentrum Informatik (FZI), Diplomarbeit, 1994
- [Cox 1991] COX, I.: Blanche—an experiment in guidance and navigation of an autonomous robot vehicle. In: *IEEE Transactions on Robotics and Automation* 7 (1991), Nr. 2, S. 193–204
- [Cozman et al. 2000] COZMAN, F. ; KROTKOV, E. ; GUESTRIN, C.: Outdoor Visual Position Estimation for Planetary Rovers. In: *Autonomous Robots* 9 (2000), S. 135–150
- [Crowley 1989] CROWLEY, J.: World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging. In: *IEEE International Conference on Robotics and Automation*, 1989, S. 674–680
- [Cruse et al. 1998] CRUSE, H. ; KINDERMANN, Th. ; SCHUMM, M. ; DEAN, J. ; SCHMITZ, J.: Walknet—a Biologically Inspired Network to Control Six-Legged Walking. In: *Neural Networks* 11 (1998), S. 1435–1447
- [Dellaert et al. 1999] DELLAERT, F. ; FOX, D. ; BURGARD, W. ; THRUN, S.: Monte Carlo Localization for Mobile Robots. In: *IEEE International Conference on Robotics and Automation*, 1999, S. 1322–1328
- [DIN 18 123] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (Hrsg.): *DIN 18 123*
- [DIN 18 300] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (Hrsg.): *DIN 18 300*

- [Dose 1995] DOSE, M.: *Wegeplanung autonomer mobiler Roboter mittels dynamischer Systeme*, Ruhr-Universität Bochum, Dissertation, 1995
- [Eldershaw und Yim 2001] ELDERSHAW, C. ; YIM, M.: Motion planning of legged vehicles in an unstructured environment. In: *IEEE International Conference on Robotics and Automation*, 2001, S. 3383–3389
- [Elfes 1989] ELFES, A.: Using Occupancy Grids for Mobile Robot Perception and Navigation. In: *Computer* 22 (1989), Nr. 6, S. 46–57
- [Engelson und McDermott 1992] ENGELSON, S. P. ; McDERMOTT, D. V.: Error Correction in Mobile Robot Map Learning. In: *IEEE International Conference on Robotics and Automation*, 1992, S. 2555–2560
- [Farrell und Barth 1998] FARRELL, J. A. ; BARTH, M.: *The Global Positioning System and Inertial Navigation*. McGraw-Hill Verlag, 1998
- [Farritor und Dubowsky 1997] FARRITOR, S. ; DUBOWSKY, S.: A Self Planning Methodology for Planetary Robotic Explorers. In: *IEEE International Conference on Robotics and Automation*, 1997, S. 499–504
- [Fox 1998] FOX, D.: *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation*. Institut für Informatik III, Universität Bonn, Dissertation, 1998
- [Fremdwörterduden 1982] BIBLIOGRAPHISCHES INSTITUT (Hrsg.): *Duden – Fremdwörterbuch*. Mannheim, 1982
- [Frik et al. 1999] FRIK, M. ; GUDDAT, M. ; KARATAS, M. ; LOSCH, C. D.: A novel approach to autonomous control of walking machines. In: *3rd International Conference on Climbing and Walking Robots*, 1999, S. 333–342
- [Frommberger 2002] FROMMBERGER, Lutz: *3D-Umweltmodellierung zur Navigationsunterstützung von Laufmaschinen im Gelände*, Forschungszentrum Informatik (FZI), Diplomarbeit, November 2002
- [Fuke und Krotkov 1996] FUKE, Y. ; KROTKOV, E.: Dead Reckoning for a Lunar Rover on uneven Terrain. In: *IEEE International Conference on Robotics and Automation*, 1996, S. 411–416
- [Gaßmann 2000] GASSMANN, B.: *Erweiterung einer modularen Laufmaschinensteuerung für unstrukturiertes Gelände*, Forschungszentrum Informatik (FZI), Diplomarbeit, 2000
- [Gaßmann et al. 2006] GASSMANN, B. ; BÄR, T. ; ZÖLLNER, J.M. ; DILLMANN, R.: Navigation of Walking Robots: Adaptation to the Terrain. In: *Proceedings of the 9th International Conference on Climbing and Walking Robots*, 2006, S. 233–239

- [Gaßmann und Berns 2002] GASSMANN, B. ; BERNs, K.: Local Navigation of LAURON III for Walking in Rough Terrain. In: *Proceedings of the 5th International Conference on Climbing and Walking Robots*, 2002, S. 509–514
- [Gaßmann et al. 2003a] GASSMANN, B. ; FROMMBERGER, L. ; DILLMANN, R. ; BERNs, K.: Improving the Walking Behaviour of a Legged Robot using a Three-Dimensional Environment Model. In: *Proceedings of the 6th International Conference on Climbing and Walking Robots*, 2003, S. 319–326
- [Gaßmann et al. 2003b] GASSMANN, B. ; FROMMBERGER, L. ; DILLMANN, R. ; BERNs, K.: Real-Time 3D Map Building for Local Navigation of a Walking Robot in Unstructured Terrain. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, S. 2185–2190
- [Gaßmann et al. 2005a] GASSMANN, B. ; HUBER, M. ; ZÖLLNER, J.M. ; DILLMANN, R.: Navigation of Walking Robots: Path Planning. In: *Proceedings of the 8th International Conference on Climbing and Walking Robots*, 2005, S. 115–122
- [Gaßmann et al. 2005b] GASSMANN, B. ; ZACHARIAS, F. ; ZÖLLNER, J.M. ; DILLMANN, R.: Localization of Walking Robots. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, S. 1483–1488
- [Gaßmann et al. 2004] GASSMANN, B. ; ZÖLLNER, J.M. ; DILLMANN, R.: Navigation of Walking Robots: Localisation by Odometry. In: *Proceedings of the 7th International Conference on Climbing and Walking Robots*, 2004, S. 953–960
- [Goel et al. 1999] GOEL, P. ; ROUMELIOTIS, S. I. ; SUKHATME, G. S.: Robust Localization Using Relative and Absolute Position Estimates. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999, S. 1134–1140
- [Grewal et al. 2001] GREWAL, M. S. ; WEILL, L. R. ; ANDREWS, A. P.: *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley – Interscience Verlag, 2001
- [Haug und Rühl 2005] HAUG, Florian ; RÜHL, Steffen: *Optimierung der Lokalisation für eine Laufmaschine*, Forschungszentrum Informatik (FZI), Studienarbeit, Dezember 2005
- [Hebert et al. 1989] HEBERT, M. ; CAILLAS, C. ; KROTKOV, E. ; KWEON, I. S. ; KANADE, T.: Terrain Mapping for a Roving Planetary Explorer. In: *IEEE International Conference on Robotics and Automation*, 1989, S. 997–1000
- [Hocaoğlu und Sanderson 2001] HOC AO ĞLU, C. ; SANDERSON, A. C.: Planning Multiple Paths with Evolutionary Speciation. In: *IEEE Transactions on Evolutionary Computation* 5 (2001), Nr. 3, S. 169–191
- [Honda 2004] HONDA: *The Honda Humanoids Robots Homepage*. <http://www.honda-robots.com>. 2004

- [Hoppen et al. 1990] HOPPEN, P. ; KNIERIEMEN, P. ; PUTTKAMER, E.: Laser-Radar Based Mapping and Navigation for an Autonomous Mobile Robot. In: *IEEE International Conference on Robotics and Automation*, 1990, S. 948–953
- [Huber 2005] HUBER, Marco: *Bahnplanung für Laufmaschinen*, Forschungszentrum Informatik (FZI), Studienarbeit, März 2005
- [Iagnemma et al. 2004] IAGNEMMA, K. ; BROOKS, C. ; DUBOWSKY, S.: Visual, Tactile, and Vibration-Based Terrain Analysis for Planetary Rovers. In: *IEEE Aerospace Conference*, 2004, S. 841–848
- [Iagnemma und Dubowsky 2002] IAGNEMMA, K. ; DUBOWSKY, S.: Terrain estimation for high-speed rough-terrain autonomous vehicle navigation. In: *SPIE Conference on Unmanned Ground Vehicle Technology IV*, 2002
- [Igarashi und Kakikura 2004] IGARASHI, H. ; KAKIKURA, M.: Path and Posture Planning for Walking Robots by Artificial Potential Field Method. In: *IEEE International Conference on Robotics and Automation*, 2004, S. 2165–2170
- [IOF Map Commission 2000] INTERNATIONAL ORIENTEERING FEDERATION – MAP COMMISSION (Hrsg.): *International Specification of Orienteering Maps*. 2000
- [Jackins und Tanimoto 1980] JACKINS, C. L. ; TANIMOTO, S. L.: Octtrees And Their Use In Representing Three-Dimensional Objects. In: *Computer Graphics and Image Processing* 14 (1980), S. 249–270
- [Jarvis 1994] JARVIS, R. A.: Distance Transform Based Collision-Free Path Planning for Robot. In: *Advanced Mobile Robots* (1994), S. 3–31
- [Jensfelt und Kristensen 2001] JENSFELT, P. ; KRISTENSEN, S.: Active global localisation for a mobile robot using multiple hypothesis tracking. In: *IEEE Transactions on Robotics and Automation* 17 (2001), Nr. 5, S. 748–760
- [Kagami et al. 1998] KAGAMI, S. ; OKADA, K. ; KABASAWA, M. ; MATSUMOTO, Y. ; KONNO, A. ; INABA, M. ; INOUE, H.: A Vision-based Legged Robot as a Research Platform. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998, S. 235–240
- [Kahlert 1995] KAHLERT, J.: *Fuzzy Control für Ingenieure*. Friedrich Vieweg & Sohn, Braunschweig, 1995
- [Kalman 1960] KALMAN, R. E.: A new approach to linear filtering and prediction problems. In: *Transactions of the ASME, Journal of Basic Engineering* (1960), Nr. 82, S. 35–45
- [Kanehiro et al. 2005] KANEHIRO, F. ; YOSHIMI, T. ; KAJITA, S. ; MORISAWA, M. ; FUJIWARA, K. ; HARADA, K. ; KANEKO, K. ; HIRUKAWA, H. ; TOMITA, F.: Whole Body Locomotion Planning of Humanoid Robots based on a 3D Grid Map. In: *IEEE International Conference on Robotics and Automation*, 2005, S. 1084–1090

- [Kaneko et al. 2004] KANEKO, K. ; KANEHIRO, F. ; KAJITA, S. ; HIRUKAWA, H. ; KAWASAKI, T. ; HIRATA, M. ; AKACHI, K. ; ISOZUMI, T.: Humanoid Robot HRP-2. In: *IEEE International Conference on Robotics and Automation*, 2004, S. 1083–1090
- [Karlsson et al. 2005] KARLSSON, N. ; BERNARDO, E. di ; OSTROWSKI, J. ; GONCALVES, L. ; PIRJANIAN, P. ; MUNICH, M.E.: The vSLAM Algorithm for Robust Localization and Mapping. In: *IEEE International Conference on Robotics and Automation*, 2005, S. 24–29
- [Kau et al. 1989] KAU, C.-C. ; OLSON, K. W. ; RIBBLE, E. A. ; KLEIN, C. A.: Design and Implementation of a Vision Processing System for a Walking Machine. In: *IEEE Transactions on Industrial Electronics* 36 (1989), Nr. 1, S. 25–33
- [Kepplin und Berns 1999] KEPPLIN, V. ; BERNS, K.: A Concept for Walking Behaviour in Rough Terrain. In: *3rd International Conference on Climbing and Walking Robots*, 1999, S. 509–516
- [Khatib 1986] KHATIB, O.: Real-time obstacle avoidance for manipulators and mobile robots. In: *The International Journal of Robotics Research* 5 (1986), Nr. 1, S. 90–99
- [Komsuoglu et al. 2001] KOMSUOGLU, H. ; MCMORDIE, D. ; SARANLI, U. ; MOORE, N. ; BUEHLER, M. ; KODITSCHKEK, D. E.: Proprioception Based Behavioural Advances in a Hexapod Robot. In: *IEEE International Conference on Robotics and Automation*, 2001, S. 3650–3655
- [Konno et al. 2002] KONNO, A. ; KATO, N. ; MITSUYA, Y. ; UCHIYAMA, M.: Design and Development of the Quadrupedal Research Platform JROB-2. In: *IEEE International Conference on Robotics and Automation*, 2002, S. 1056–1061
- [Kowoma 2004] KOWOMA: *Erreichbare Genauigkeit bei GPS-Positionsbestimmungen*. <http://www.kowoma.de/gps/Genauigkeit.htm>. 2004
- [Kruse et al. 1995] KRUSE, E. ; GUTSCHE, R. ; WAHL, F. M.: Inkrementelle sensorbasierte Erzeugung eines Umweltmodells mit Hilfe von Bewertungsfunktionen im Konfigurationsraum. In: DILLMANN, Rüdiger (Hrsg.) ; REMBOLD, Ulrich (Hrsg.) ; LÜTH, Tim (Hrsg.): *Autonome Mobile Systeme 1995*. Springer Verlag Heidelberg Berlin, 1995 (Informatik aktuell), S. 120–131
- [Kuipers und Byun 1991] KUIPERS, B. J. ; BYUN, Y.-T.: A Robot Exploration and Mapping Strategy based on a Semantic Hierarchy of Spatial Representations. In: *Journal of Robotics and Autonomous Systems* 8 (1991), S. 47–63
- [Kumagai und Emura 2000] KUMAGAI, M. ; EMURA, T.: Vision Based Walking of Human Type Biped Robot on Undulating Ground. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000, S. 1352–1357

- [Laird et al. 1986] LAIRD, J. E. ; ROSENBLOOM, P. S. ; NEWELL, A.: Chunking in SOAR: The anatomy of a general learning mechanism. In: *Machine Learning* 1 (1986), Nr. 1, S. 11–46
- [Lamon und Siegwart 2003] LAMON, P. ; SIEGWART, R.: 3D-Odometry for rough terrain – Towards real 3D navigation. In: *IEEE International Conference on Robotics and Automation*, 2003, S. 440–445
- [Latombe 1991] LATOMBE, J.: *Robot Motion Planning*. Kluwer Academic Publishers, 1991
- [Launay et al. 2002] LAUNAY, F. ; OHYA, A. ; YUTA, S. ichi: A Corridors Lights based Navigation System including Path Definition using a Topologically Corrected Map for Indoor Mobile Robots. In: *IEEE International Conference on Robotics and Automation (ICRA)*, 2002, S. 3918–3923
- [Lee und Orin 1988] LEE, W.-J. ; ORIN, D. E.: The Kinematics of Motion Planning for Multilegged Vehicles Over Uneven Terrain. In: *IEEE Journal of Robotics and Automation* 4 (1988), Nr. 2, S. 204–212
- [Leonard und Durant-White 1991a] LEONARD, J. J. ; DURANT-WHITE, H. F.: Mobile Robot Localization by Tracking Geometric Beacons. In: *IEEE Transactions on Robotics and Automation* 7 (1991), Nr. 3, S. 376–382
- [Leonard und Durant-White 1991b] LEONARD, J. J. ; DURANT-WHITE, H. F.: Simultaneous Map Building and Localization for an Autonomous Mobile Robot. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1991, S. 1442–1447
- [Lin et al. 2005] LIN, P.-C. ; KOMSUOGLU, H. ; KODITSCHKEK, D. E.: A leg configuration measurement system for full-body pose estimates in a hexapod robot. In: *IEEE Transactions on Robotics and Automation* 21 (2005), Nr. 3, S. 411–422
- [Lorch et al. 2002] LORCH, O. ; ALBERT, A. ; DENK, J. ; GERECKE, M. ; CUPEC, R. ; SEARA, J. F. ; GERTH, W. ; SCHMIDT, G.: Experiments in Vision Guided Biped Walking. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, S. 2484–2490
- [Lozano-Pérez 1983] LOZANO-PÉREZ, T.: Spatial Planning: A configuration space approach. In: *IEEE Transactions on Computers* C-32 (1983), Nr. 2, S. 108–120
- [Mae et al. 2001] MAE, Y. ; MASUDA, T. ; ARAI, T. ; INOUE, K.: Error Analysis of Dead Reckoning of Multi-Legged Robots. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, S. 1558–1563
- [Martinelli und Siegwart 2003] MARTINELLI, A. ; SIEGWART, R.: Estimating the Odometry Error of a Mobile Robot during Navigation. In: *European Conference on Mobile Robots*, 2003

- [Maybeck 1979] MAYBECK, P. S.: *Stochastic models, Estimation and Control*. Bd. 1. Academic Press, 1979
- [McClain und Conrad 2005] McCLAIN, A. ; CONRAD, J. M.: Software Design of the Stiquito Micro Robot. In: *IEEE SoutheastCon*, 2005, S. 143–147
- [Mitchell 1997] MITCHELL, T. M.: *Machine Learning*. International Edition. McGraw-Hill, 1997
- [Molfino et al. 2005] MOLFINO, R. ; ARMADA, M. ; CEPOLINA, F. ; ZOPPI, M.: Roboclimber, the 3 Tons Spider. In: *Industrial Robot Journal* 32 (2005), Nr. 2
- [Moore und Atkeson 1995] MOORE, A. W. ; ATKESON, C. G.: The Parti-Game Algorithm For Variable Resolution Reinforcement Learning In Multidimensional State-Spaces. In: *Machine Learning* 21 (1995)
- [Moravec 1988] MORAVEC, B. P.: Certainty Grids for Sensor Fusion in Mobile Robots. In: *AI Magazine* 9 (1988), Nr. 2, S. 61–77
- [Murphy et al. 2002] MURPHY, K. ; ABRAMS, M. ; BALAKIRSKY, S. ; CHANG, T. ; HONG, T. ; LACAZE, A. ; LEGOWIK, S.: Intelligent Control for Off-Road Driving. In: *First International NAISO Congress on Autonomous Intelligent System*, 2002
- [Nachtigall und Büchel 2000] NACHTIGALL, W. ; BÜCHEL, K.: *Das große Buch der Bionik – Neue Technologien nach dem Vorbild der Natur*. DVA, 2000
- [Nebot et al. 1997] NEBOT, E. ; SUKKARIEH, S. ; DURRANT-WHYTE, H.: Inertial Navigation aided with GPS Information. In: *Fourth Annual Conference on Mechatronics and Machine Vision in Practice*, 1997, S. 169–174
- [Okada et al. 2005] OKADA, K. ; OGURA, T. ; HANEDA, A. ; INABA, M.: Autonomous 3D walking system for a humanoid robot based on visual step recognition and 3D foot step planner. In: *IEEE International Conference on Robotics and Automation*, 2005, S. 625–630
- [Patel et al. 2005] PATEL, K. ; MACKLEM, W. ; THRUN, S. ; MONTEMERLO, M.: Active Sensing for High-Speed Offroad Driving. In: *IEEE International Conference on Robotics and Automation*, 2005, S. 3162–3168
- [Prassler et al. 2000] PRASSLER, E. ; SCHWAMMKRUG, D. ; ROHRMOSER, B. ; SCHMIDL, G.: A Robotic Road Sweeper. In: *IEEE International Conference on Robotics and Automation*, 2000, S. 2364–2369
- [Proyas 2004] PROYAS, Alex: *I, Robot*. 20th Century Fox. 2004
- [Raibert 1985] RAIBERT, M. H.: *Legged Robots That Balance*. MIT Press, 1985
- [Rascke und Borenstein 1990] RASCHE, U. ; BORENSTEIN, J.: A Comparison of Grid-type Map-building techniques by Index of Performance. In: *IEEE International Conference on Robotics and Automation*, 1990, S. 1021–1026

- [Regenstein und Dillmann 2003] REGENSTEIN, K. ; DILLMANN, R.: Design of an open hardware architecture for the humanoid robot ARMAR. In: *International Conference on Humanoid Robots*, 2003
- [Roston und Krotkov 1992] ROSTON, G. P. ; KROTKOV, E. P.: Dead Reckoning Navigation for Walking Robots. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1992, S. 607–612
- [Russell und Norvig 2004] RUSSELL, S. ; NORVIG, P.: *Künstliche Intelligenz – Ein moderner Ansatz*. 2. Auflage. Pearson Studium, 2004
- [de Santos et al. 2003] SANTOS, P.G. de ; GALVEZ, J.A. ; ESTREMER, J. ; GARCIA, E.: SIL04: a true walking robot for the comparative study of walking machine techniques. In: *IEEE Robotics and Automation Magazine* 10 (2003), Nr. 4, S. 23–32
- [Scholl et al. 2001] SCHOLL, K.-U. ; ALBIEZ, J. ; GASSMANN, B.: MCA - An Expandable Modular Controller Architecture. In: *In proceedings of the 4th Linux Real Time Workshop*, 2001
- [Seara et al. 2002] SEARA, J. F. ; STROBL, K. H. ; SCHMIDT, G.: Information Management for Gaze Control in Vision Guided Biped Walking. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, S. 31–36
- [Selkänaho et al. 2001] SELKÄINAHO, J. ; HALME, A. ; PAANAJÄRVI, J.: Navigation System Of An Outdoor Service Robot With Hybrid Locomotion System. In: *International Conference on Field and Service Robotics*, 2001, S. 79–83
- [Seraji und Howard 2002] SERAJI, H. ; HOWARD, A.: Behavior-based robot navigation on challenging terrain: A fuzzy logic approach. In: *IEEE Transactions on Robotics and Automation* 18 (2002), Nr. 3, S. 308–321
- [Shatkay und Kaelbling 1997] SHATKAY, H. ; KAEHLING, L. P.: Learning Topological Maps with Weak Local Odometric Information. In: *Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, 1997, S. 920–927
- [Shneider et al. 2004] SHNEIER, M. ; HONG, T. ; CHANG, T. ; SCOTT, H. ; LEGOWIK, S. ; CHEOK, G. ; GIAUQUE, C. ; GILSINN, D. ; WITZGALL, C.: Terrain Characterization for TRL-6 Evaluation of an Unmanned Ground Vehicle / National Institute of Standards and Technology. 2004. – Forschungsbericht.
- [Simhons und Dudek 1998] SIMHONS, S. ; DUDEK, G.: A Global Topological Map formed by Local Metric Maps. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998, S. 1708–1714
- [Simmons und Krotkov 1991] SIMMONS, R. ; KROTKOV, E.: An Integrated Walking System for the Ambler Planetary Rover. In: *IEEE International Conference on Robotics and Automation*, 1991, S. 2086–2091

- [Skaff et al. 2003] SKAFF, S. ; KANTOR, G. ; MAIWAND, D. ; RIZZI, A. A.: Inertial Navigation and Visual Line Following for a Dynamical Hexapod Robot. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, S. 1808–1813
- [Spandl 1991] SPANDL, H.: *Lernverfahren zur Unterstützung der Routenplanung für einen mobilen Roboter*, Universität Karlsruhe, Dissertation, 1991
- [Spenneberger und Kirchner 2002] SPENNEBERGER, D. ; KIRCHNER, F.: SCORPION: A Biomimetic Walking Robot. In: *Robotik 2002*, 2002 (VDI-Bericht), S. 677–682
- [Spenneberger et al. 2004] SPENNEBERGER, D. ; MCCULLOUGH, K. ; KIRCHNER, F.: Stability of Walking in a Multilegged Robot Suffering Leg Loss. In: *IEEE International Conference on Robotics and Automation*, 2004, S. 2159–2164
- [Stein und Medioni 1995] STEIN, F. ; MEDIONI, G.: Map-based Localization Using the Panoramic Horizon. In: *IEEE Transactions on Robotics and Automation* 11 (1995), Nr. 6, S. 892–896
- [Sukkarieh et al. 1999] SUKKARIEH, S. ; NEBOT, E. ; DURRANT-WHYTE, H. F.: A High Integrity IMU/GPS Navigation Loop for Autonomous Land Vehicle Applications. In: *IEEE Transactions on Robotics and Automation* 15 (1999), Nr. 3, S. 571–578
- [Tagscherer 1994] TAGSCHERER, Michael: *Adaptive Umweltklassifikation für eine Laufmaschine*, Forschungszentrum Informatik (FZI), Diplomarbeit, 1994
- [Thrun 1998] THRUN, S.: Learning Maps for Indoor Mobile Robot Navigation. In: *Artificial Intelligence* 99 (1998), S. 21–71
- [Thrun und Bücken 1996] THRUN, S. ; BÜCKEN, A.: Integrating Grid-Based and Topological Maps for Mobile Robot Navigation. In: *Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, 1996
- [Thrun et al. 1998] THRUN, S. ; GUTMANN, J.-S. ; FOX, D. ; BURGARD, W. ; KUIPERS, B. J.: Integrating Topological and Metric Maps for Mobile Robot Navigation: A Statistical Approach. In: *AAAI Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 1998, S. 989–995
- [Titterton und Weston 1997] TITTERTON, D. H. ; WESTON, J. L.: *Strapdown Inertial Navigation Technology*. Peter Peregrinus Ltd./IEE, 1997
- [Todt und Torras 2000] TODT, E. ; TORRAS, C.: Detection of natural landmarks through multiscale opponent features. In: *15th International Conference on Pattern Recognition*, 2000, S. 976–979
- [Tsumura 1986] TSUMURA, T.: Survey of Automated Guided Vehicles in Japanese Factory. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1986, S. 1329–1334

- [Vandapel et al. 2004] VANDAPEL, N. ; HUBER, D.F. ; KAPURIA, A. ; HEBERT, M.: Natural Terrain Classification using 3-D Ladar Data. In: *IEEE International Conference on Robotics and Automation*, 2004, S. 5117–5122
- [Volpe 1999] VOLPE, R.: Mars Rover Navigation Results Using Sun Sensor Heading Determination. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999, S. 460–467
- [Volpe 2005] VOLPE, R.: Rover Technology Development and Mission Infusion Beyond MER. In: *IEEE Conference Aerospace*, 2005, S. 1–11
- [Volpe et al. 2000] VOLPE, R. ; ESTLIN, T. A. ; LAUBACH, S. L. ; OLSON, C. F. ; BALARAM, J.: Enhanced Mars Rover Navigation Techniques. In: *IEEE International Conference on Robotics and Automation*, 2000, S. 926–931
- [Weidemann et al. 1994] WEIDEMANN, H. J. ; PFEIFFER, F. ; ELZE, J.: The Six-Legged TUM Walking Robot. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1994, S. 1026–1033
- [Weingarten et al. 2004] WEINGARTEN, J.D. ; LOPES, G.A.D. ; BUEHLER, M. ; GROFF, R.E. ; KODITSCHKE, D.E.: Automated gait adaptation for legged robots. In: *IEEE International Conference on Robotics and Automation*, 2004, S. 2153–2158
- [Welch und Bishop 2001] WELCH, G. ; BISHOP, G.: *An Introduction to the Kalman Filter*. Workshop at the SIGGRAPH, Course 8. 2001
- [Wendel 2003] WENDEL, J.: *Entwurfs- und Analysemethoden Integrierter Navigationssysteme*. Fakultät für Elektrotechnik und Informationstechnik, Universität Karlsruhe, Dissertation, 2003
- [Wettergreen et al. 1990] WETTERGREEN, D. ; THOMAS, H. ; THORPE, C.: Planning Strategies for the Ambler Walking Robot. In: *IEEE International Conference on Systems Engineering*, 1990, S. 198–203
- [White 2002] WHITE, R.: *The Kalman Filter*. Air Force Research Laboratory's Office of Scientific Research <http://www.afrlhorizons.com/Briefs/Mar02/OSR0106.html>. 2002
- [Wiedemann 2003] WIEDEMANN, Adiran: *Odometrie von Laufmaschinen*, Forschungszentrum Informatik (FZI), Studienarbeit, Dezember 2003
- [Wikipedia 2004] WIKIPEDIA: *Die freie Enzyklopädie*. <http://de.wikipedia.org>. 2004
- [Wöhe 2000] WÖHE, Günter: *Einführung in die Allgemeine Betriebswirtschaftslehre*. Verlag Franz Vahlen, 2000

- [Ylönen und Halme 2002] YLÖNEN, S. J. ; HALME, A. J.: WorkPartner – Centaur Like Service Robot. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, S. 727–732
- [Zacharias 2004] ZACHARIAS, Franziska: *Lokalisation von Laufmaschinen im Gelände*, Forschungszentrum Informatik (FZI), Diplomarbeit, Oktober 2004
- [Zadeh 1965] ZADEH, L. A.: Fuzzy Sets. In: *Information and Control* 8 (1965), S. 338–353
- [Zadeh 1973] ZADEH, L. A.: Outline of a New Approach to The Analysis of Complex Systems And Decision Processes. In: *IEEE Transactions on Systems, Man, and Cybernetics* 2 (1973), S. 28–44
- [Zhang und Faugeras 1992] ZHANG, Z. ; FAUGERAS, O. D.: Estimation of Displacement From Two 3D Frames Obtained From Stereo. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), Nr. 12, S. 1141–1156

Insbesondere in natürlichem Gelände ist Laufen oft flexibler und effizienter als radgetriebene Fahrzeuge. Um einer Laufmaschine eine zielgerichtete, autonome Fortbewegung in Umgebungen wie beispielsweise Vulkan- oder Waldgebieten zu ermöglichen, ist es erforderlich, Navigationsmethoden einzusetzen, die den Standort des Roboters erfassen und unter Berücksichtigung der Umgebungsmerkmale einen Weg zum Ziel berechnen sowie schließlich beschreiten.

In der vorliegenden Arbeit werden Navigationsstrategien erforscht, die es einer Laufmaschine im Gelände ermöglichen, ein vorgegebenes Ziel autonom zu erreichen. Hierbei werden die Teilbereiche der Navigation, Lokalisation, Umweltmodellierung, Bahnplanung sowie Adaption an das Gelände unter der Berücksichtigung der speziellen Anforderungen und Randbedingungen von Laufmaschinen eingehend untersucht. Auf der Grundlage bestehender Methoden wurden in diesen Bereichen neuartige Algorithmen und Verfahren vorgeschlagen und anhand eines Demonstrators, der sechsbeinigen Laufmaschine LAURON 4, evaluiert.